Elements of Learning Algorithms for Natural Scene Understanding

constructing software by expressing bias and loss

Siniša Šegvić
UniZg-FER D307

Agenda

- Introduction
- Scale invariance and equivariance
- Ladder-style upsampling
- Dense connectivity
- Efficient inference
- Open-set performance
- Challenges and opportunities
- Conclusions

Introduction: change of tide

Deep learning caused profound changes into computer vision methodology

Many of our beloved methods had rapidly fell out of luck, eg:

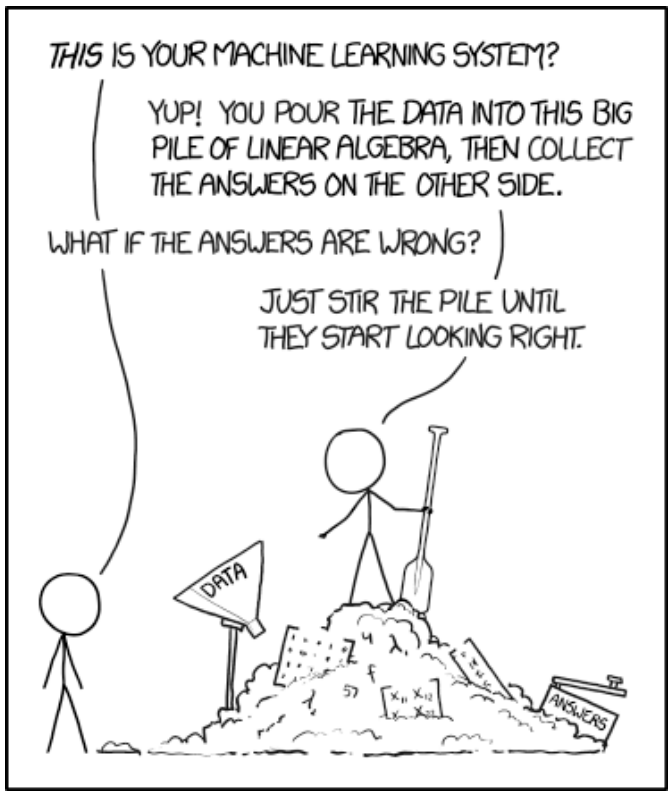
- handcrafted features (SIFT)
- handcrafted kernels (RBF)
- convex optimization (SVM)
- shallow embeddings (BoW, Fisher)

Shift from software-centric towards data-centric paradigm?

Introduction: brave new world

A popular view on contemporary computer vision development:

collect data, train a black-box model, repeat.



[xkcd1838]

It may appear as if we act as data janitors instead of programmers, research engineers or researchers.

Introduction: red or blue?

However, popular views often miss the point.

[The Matrix]

Our task is to control the caos by setting the rules of the game:

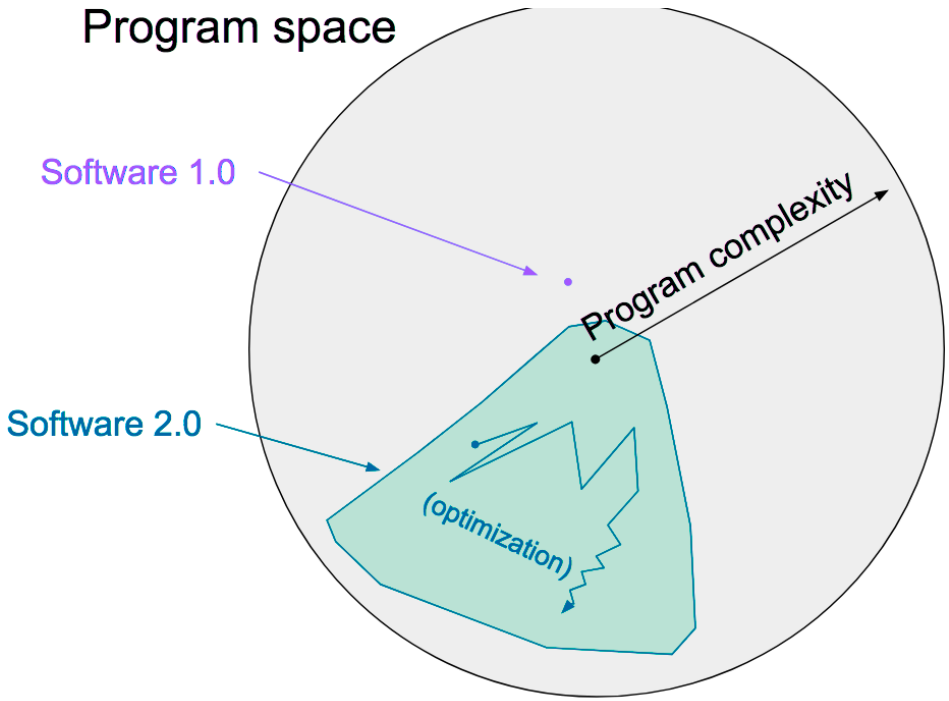
- introduce preference towards some solutions (inductive bias)
- allow training on weak labels and unlabeled data
- encourage robustness to domain shift and anomalous input
- ...

Our presence is still required in the driver's seat.

Introduction: red or blue? (2)

The rules of the game outline a large class of solutions (green region)

The optimization arranges implementation details according to data



[karpathy17medium]

Much more powerful than classic software development (violet)!

Introduction: inductive bias

Inductive bias --- preference of a learning algorithm towards a class of solutions:

- fundamental concept of machine learning
- it defines generalization from the training data to the test data

Learning without bias is futile [bašić11su].

Constructing inductive bias an important technique for designing deep learning algorithms.

Introduction: convolutional models

Convolutional layers express the following inductive bias:

- a translated image gives rise to translated activations (translational equivariance)

Inductive bias of pooling layers:

- activations do not depend on object location in the image (translational invariance)

These two pieces of inductive bias are the reason why convolutional models outperform fully connected models

Introduction: recurrent layers

A recurrent layer updates the latent state h sequentially, with respect to each token x of the input sequence:

$$h_i = f_\theta(h_{i-1}, x_i)$$

Such layers express the following inductive bias:

- all tokens are processed according to same parameters θ
- influence of a particular token does not depend on its position in the sequence

A more abstract formulation [abnar20github] applicable even when we use positional embedding:

- input tokens are processed sequentially
- there is no direct access to the past tokens

Scale invariance: something's fishy

There are infinitely many useful pieces of inductive bias:

- the list is limited only by our imagination.

For instance, note that convolutional layers are not scale-equivariant:

- a scaled image results in a different convolutional representation
- there is no deterministic relation between convolutional representations of scaled objects (??!)

[cordts16cvpr]

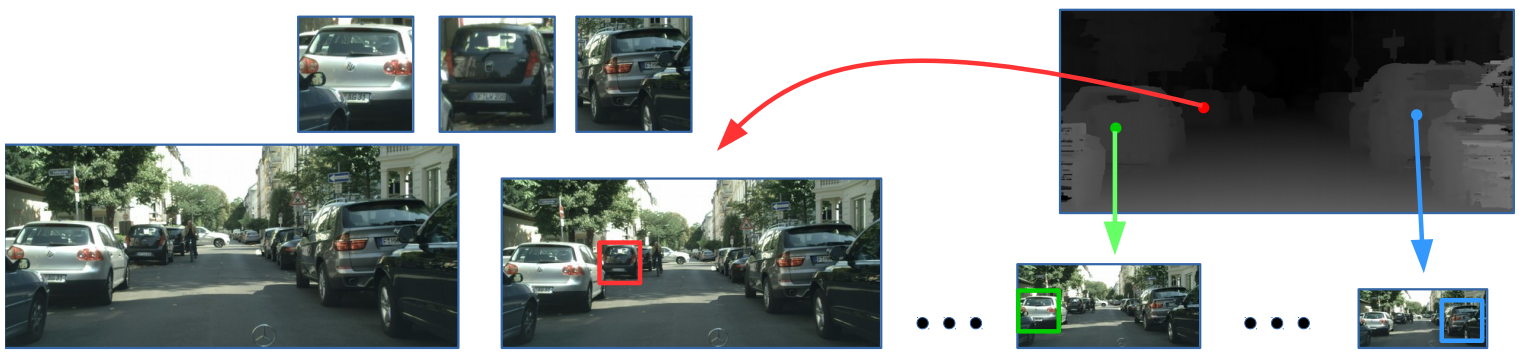
Scale invariance: hand-crafted perspective

This state of affairs does not feel right (to us at least):

- a model learns (per-class) perspective by heart?
- especially in real-time constraints where capacity is scarce

We have addressed this by promoting equivariance to scale:

- analyze each pixel at a scale which matches its stereo depth
- assemble scale-invariant representation through scale selection



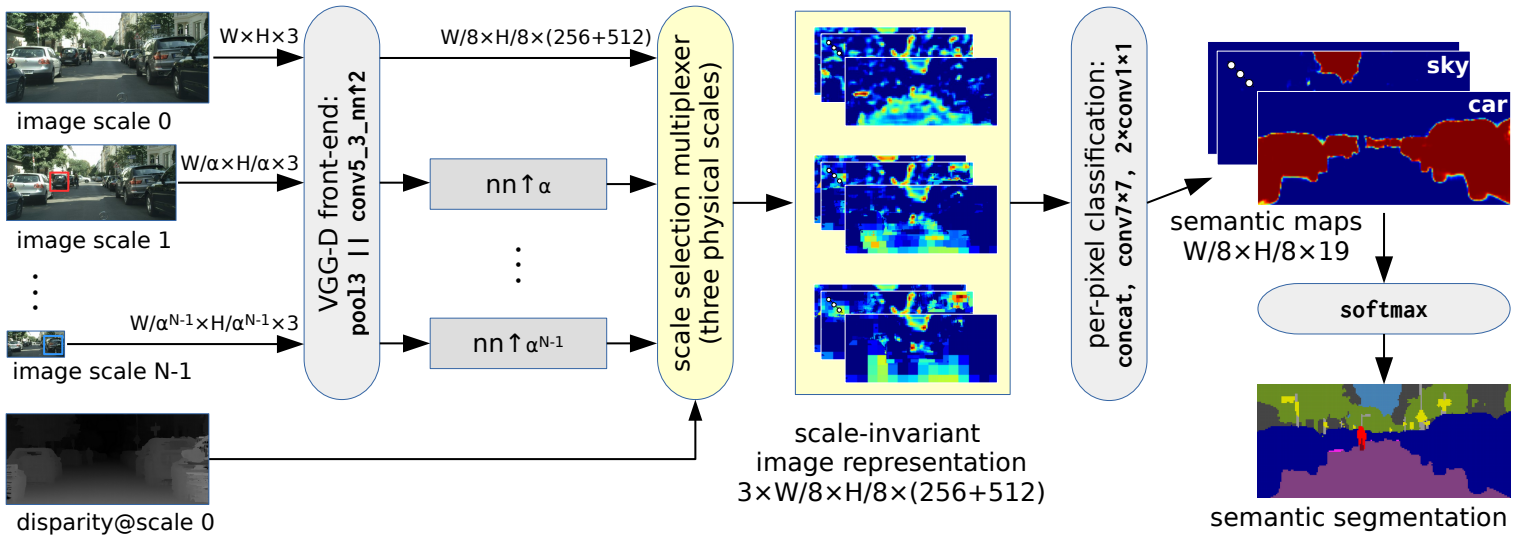
[kreso21phd]

Scale invariance: hand-crafted perspective (2)

Efficient (GPU friendly) implementation:

- apply a shared backbone across a resolution pyramid [farabet13pami]
- use pixel-level depth information to pick appropriate scale

This presents all parts of the scene as if they were filmed from n=3 canonical distances.



[kreso16gcpr]

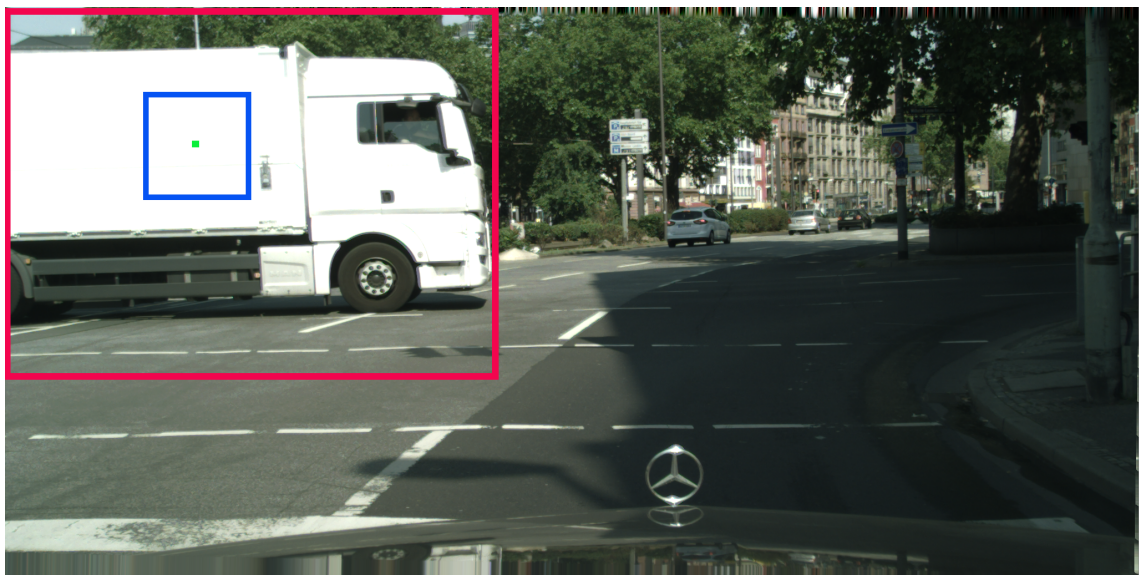
Scale invariance: hand-crafted perspective (3)

Our inductive bias contributed 3pp mIoU (Cityscapes val) over a baseline with three fixed scales and no scale selection.

We noticed most improvement at rare classes and large objects:

- this suggests that our model had insufficient receptive field
- likely caused by pre-training on 224x224 ImageNet images.

Size of the receptive field is critical for recognition of large objects:



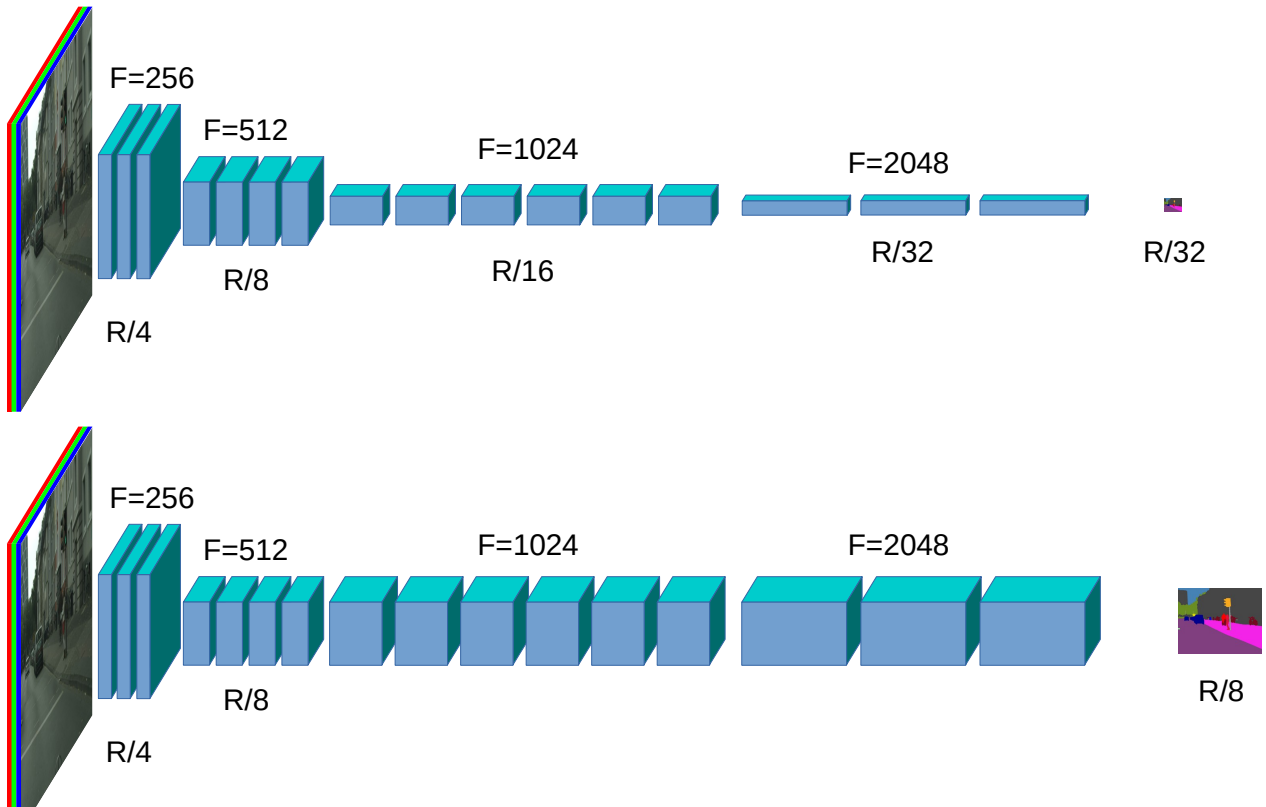
[kreso16gcpr]

Ladder-style: dilation?

Our scale invariant model was ranked #3 on Cityscapes test (2016).

However, it could not compete with later submissions which combined large convolutional backbones with dilated convolutions.

- dilated models reduce subsampling and retain pre-training
- increased computational strain and memory footprint (blue bricks)



Ladder-style: brute force or efficient?

Unfortunately (or fortunately) we could not afford dilated models:

- huge training footprint, huge computational power
- our competitors trained on 4×Titan GTX
  - unavailable in Croatia, expensive
- it makes no sense to compete from a handcapped position

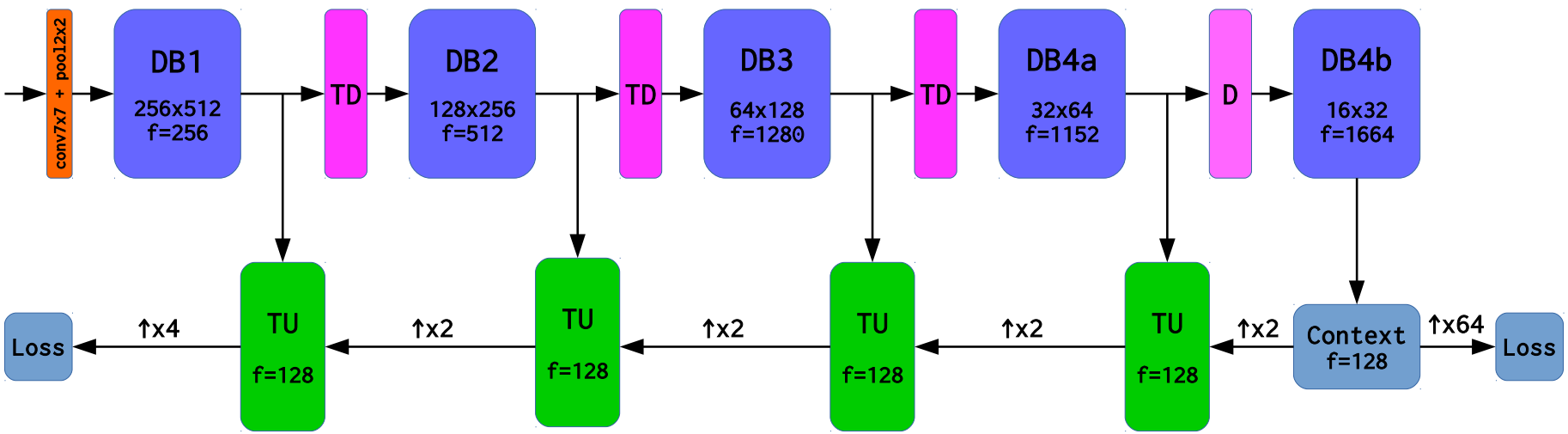
Instead, we chose to compete by making our models more efficient:

- increase the subsampling instead of increasing the computations
- very attractive due to opportunity to address real-time applications
  - robotics, driver assistance, mobile phones

Ladder-style: concept

We therefore complemented ImageNet-pretrained convolutional backbone with lightweight ladder-style upsampling:

- most layers and most capacity assigned to the backbone
- context recognition module increases the receptive field of the most compressed representation
- ladder-style upsampling blends low-resolution semantics with high-resolution details

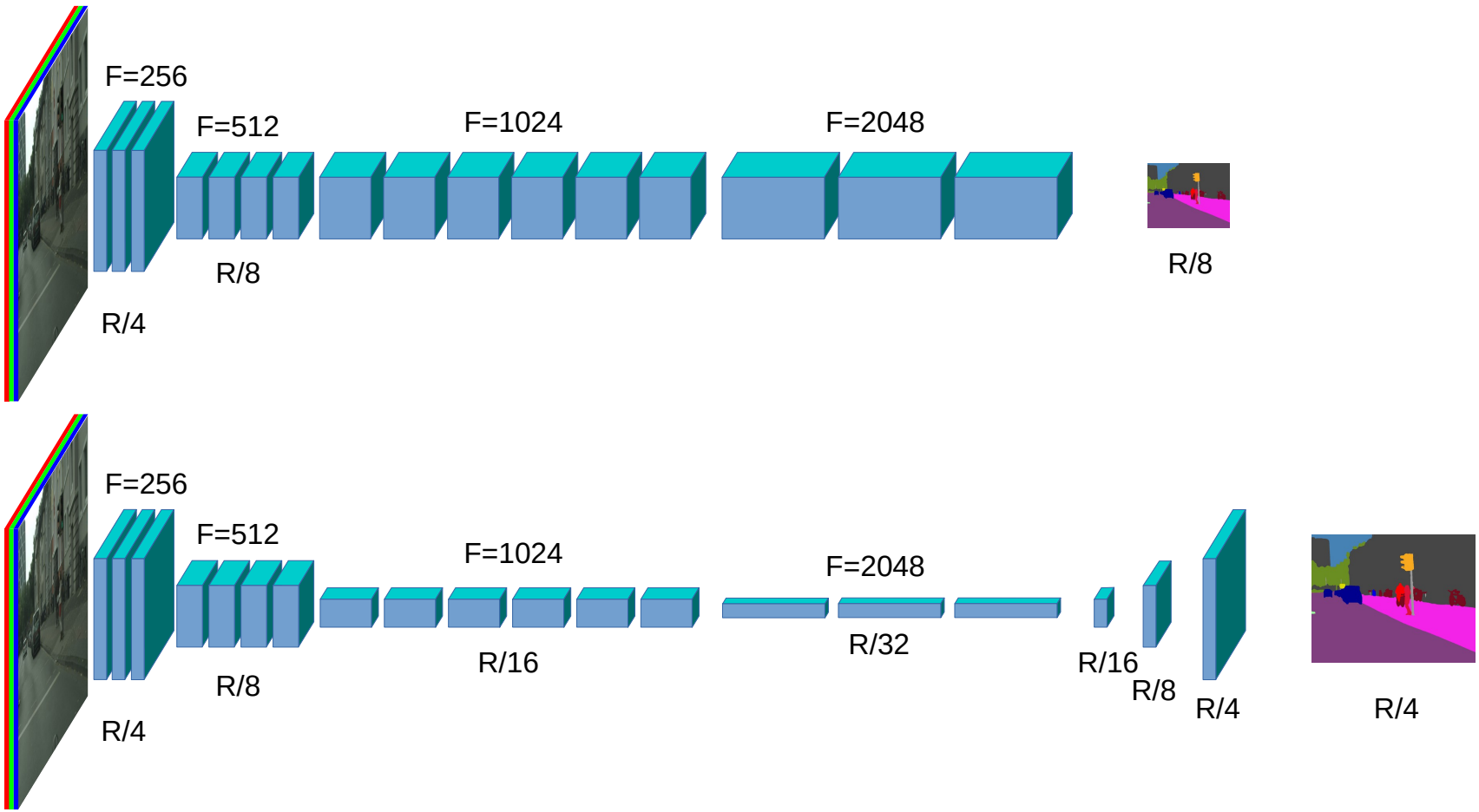


[kreso17iccvw]

Ladder-style: bias

Inductive bias of ladder-style upsampling:

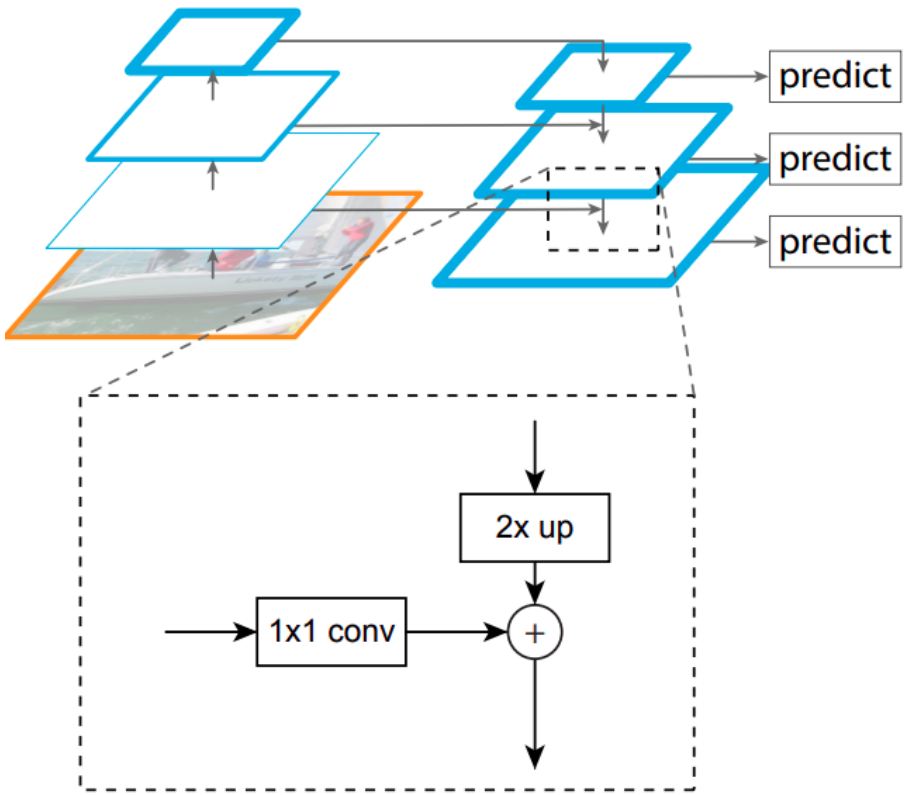
- recognition receives more capacity than border refinement
- bonus: much less computations (blue bricks) than dilated models



Ladder-style: related work

A similar architecture has been proposed in the paper on feature pyramid networks (2017, the same year as Ladder DenseNet):

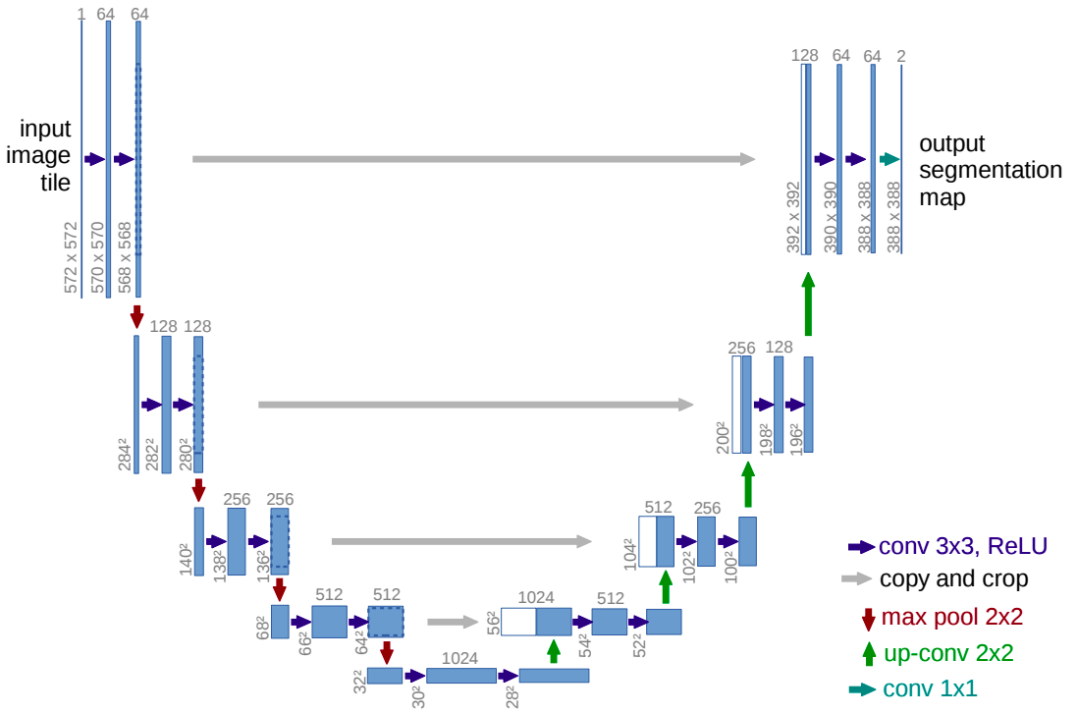
- they consider only object detection and they do not address receptive field of dense predictions



[lin17cvpr]

Ladder-style: related work (2)

Lateral skip connections have also been used in the UNet architecture:



[ronneberger15miccai]

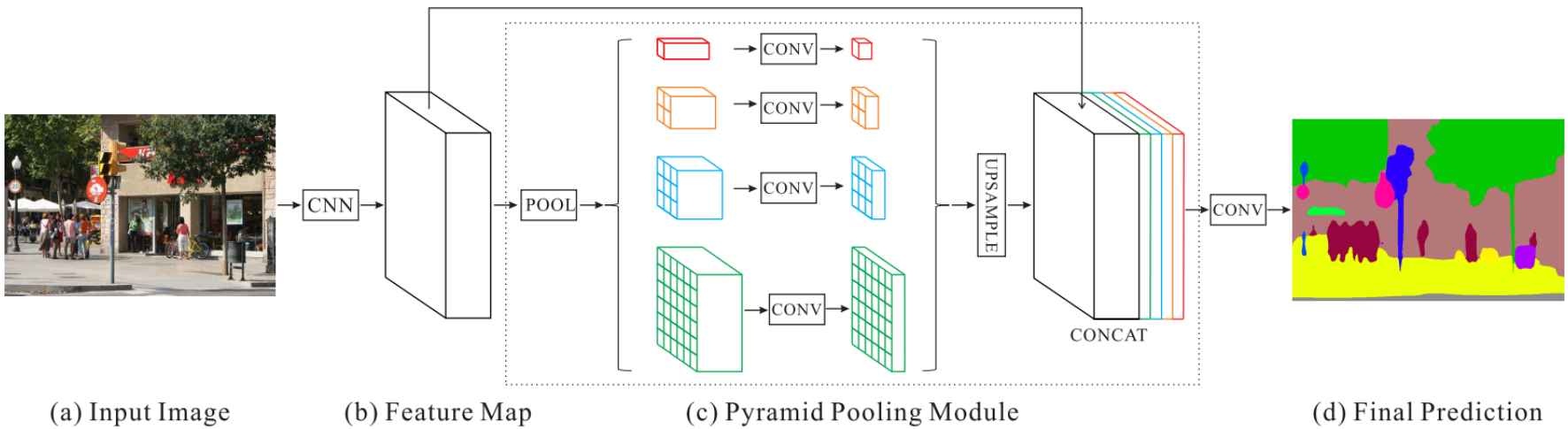
Our architecture outperforms UNets due to following advantages:

- asymetric design: +generalization, -computations
- increased receptive field due to context module
- standard recognition backbone allows pre-training

Ladder-style: pyramid pooling

Convolutional pyramid pooling [zhao17cvpr]:

- augments each feature with a context descriptor
- context descriptors are recovered through multi-grid pooling and bilinear upsampling



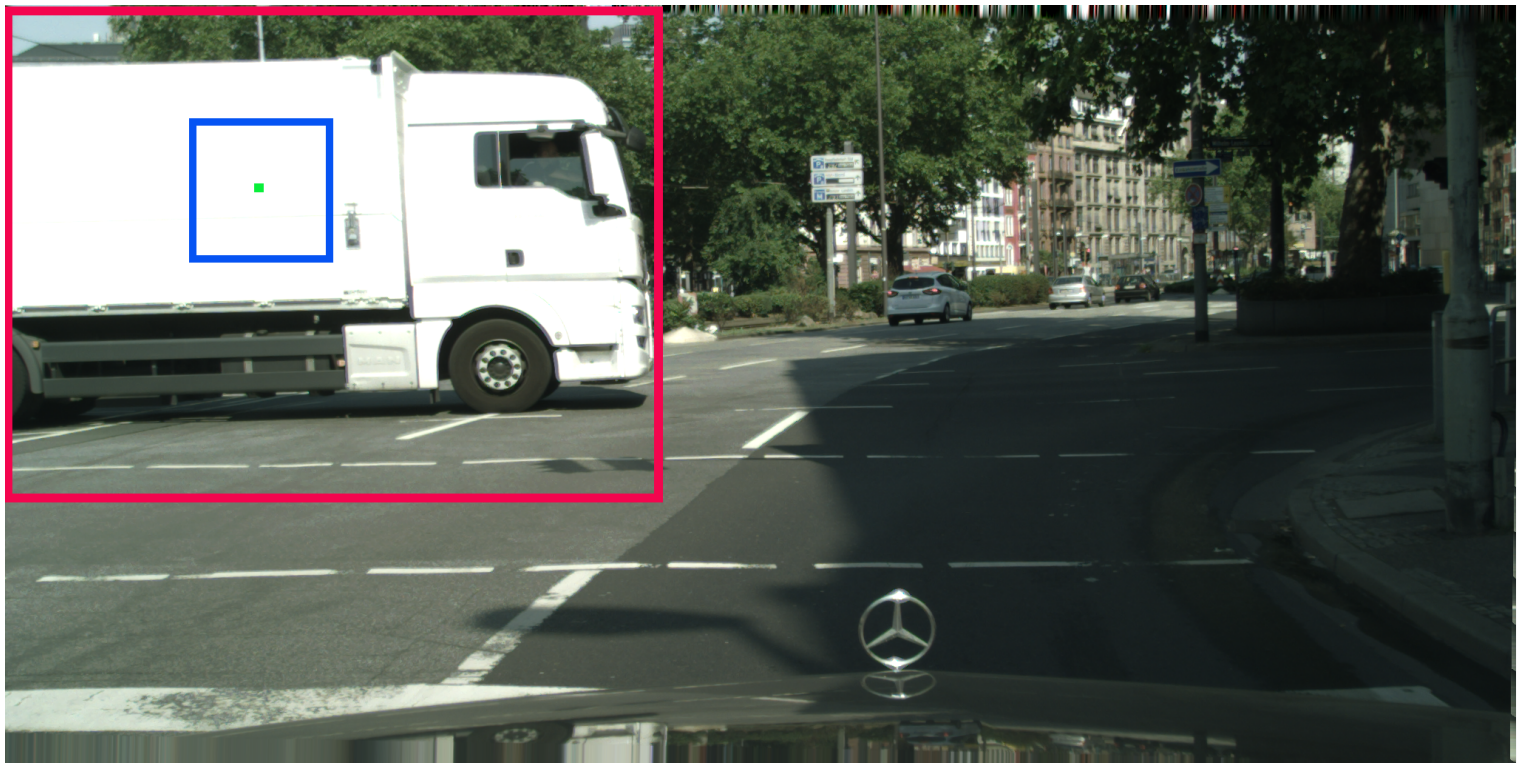
[zhao17cvpr]

Previous uses of pyramid pooling:

- augmenting image-wide representations in convolutional [he15pami] and classical BoW models [lazebnik06cvpr].

Ladder-style: pyramid pooling (2)

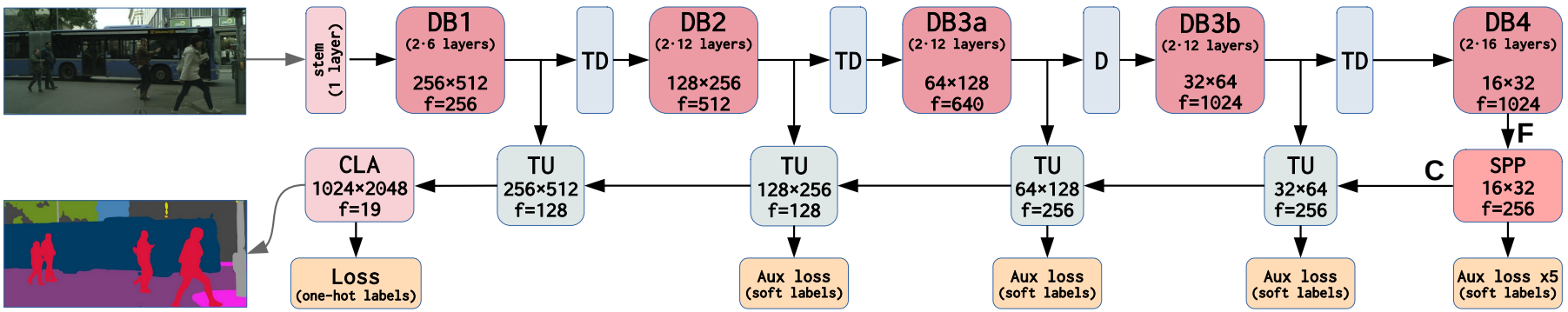
Pyramid pooling allows the model to recognize pixels on smooth surfaces by relying on context:



[kreso16gcpr]

Ladder-style: pyramid pooling (3)

Different than in [zhao17cvpr] we apply convolutional pooling at R/32 (before ladder-style upsampling):



[kreso18arxiv]

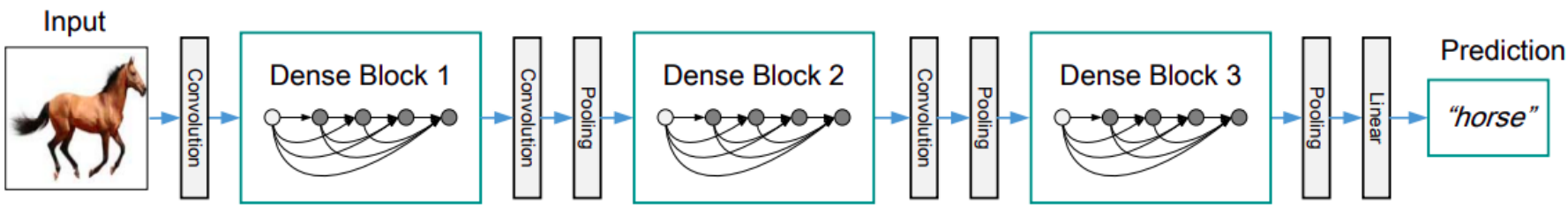
This achieves similar (or better) effects with much less computational power [kreso21tits].

Dense connectivity: approach

DenseNet architecture [huang17cvpr] has several advantages which make it our default in many different tasks.

A DenseNet model consists of 3-5 procesing blocks:

- multi-unit convolutional modules (6-100+ convolutions)
- all these convolutions operate at the same resolution
- other architectures (AlexNet, VGG, ResNet) have similar structure

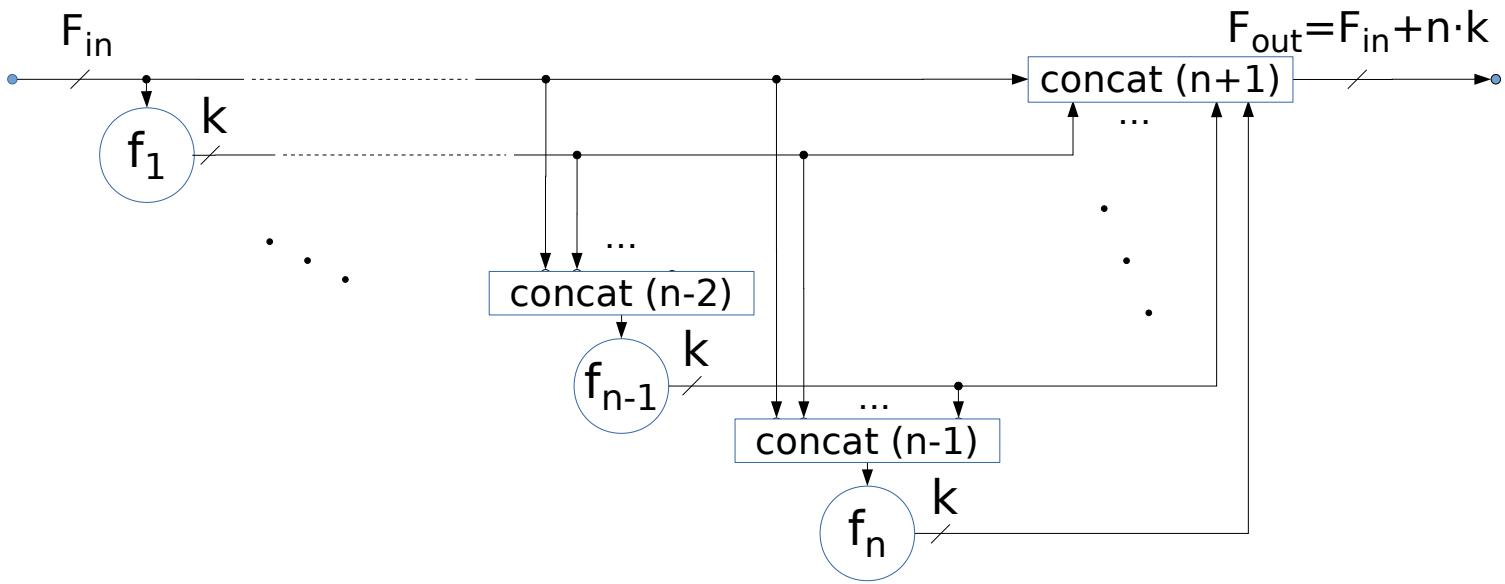


[huang17cvpr]

Dense connectivity: details

A DenseNet block relies on dense connectivity and concatenations:

- each unit operates on all preceeding units from the same block
- the output of the block is a concatenation of all units.



[kreso21phd]

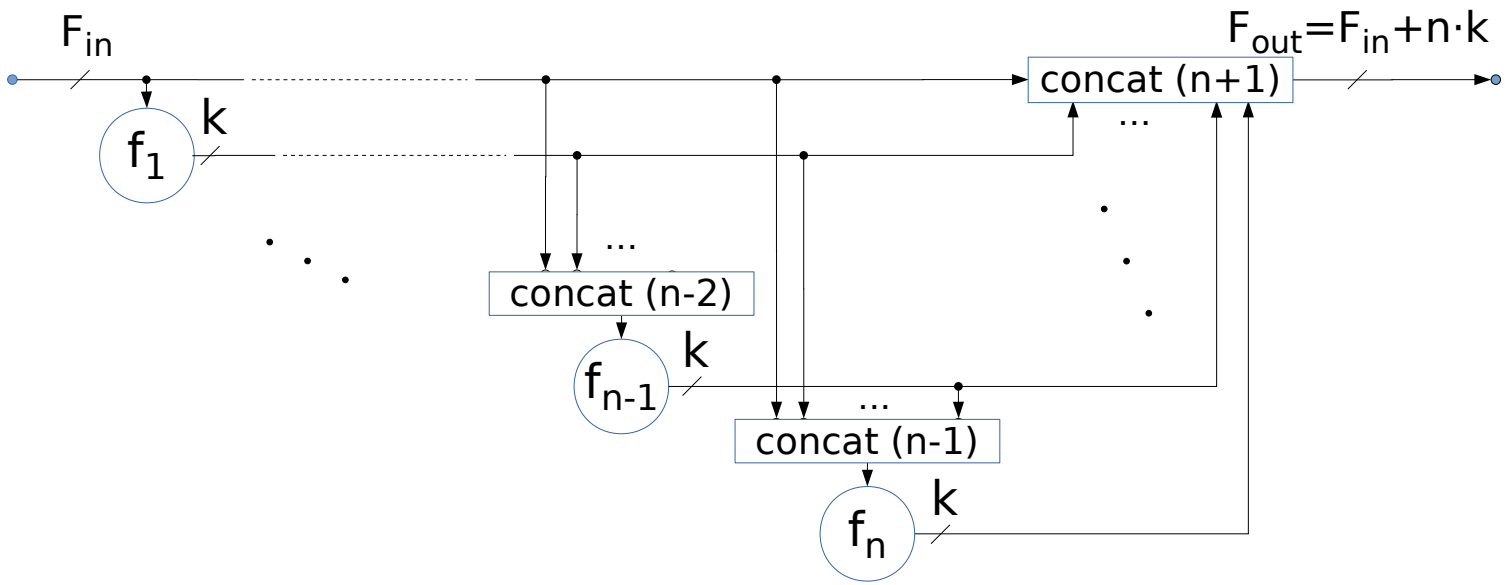
The solution is expressed through features of different complexity

- this inductive bias makes DenseNets very efficient

Dense connectivity: footprint - theory

DenseNets have a great potential to reduce the training footprint:

- backprop caches inputs for all layers with multiplicative parameters
- these inputs could be assembled by concatenating f_1-f_n.



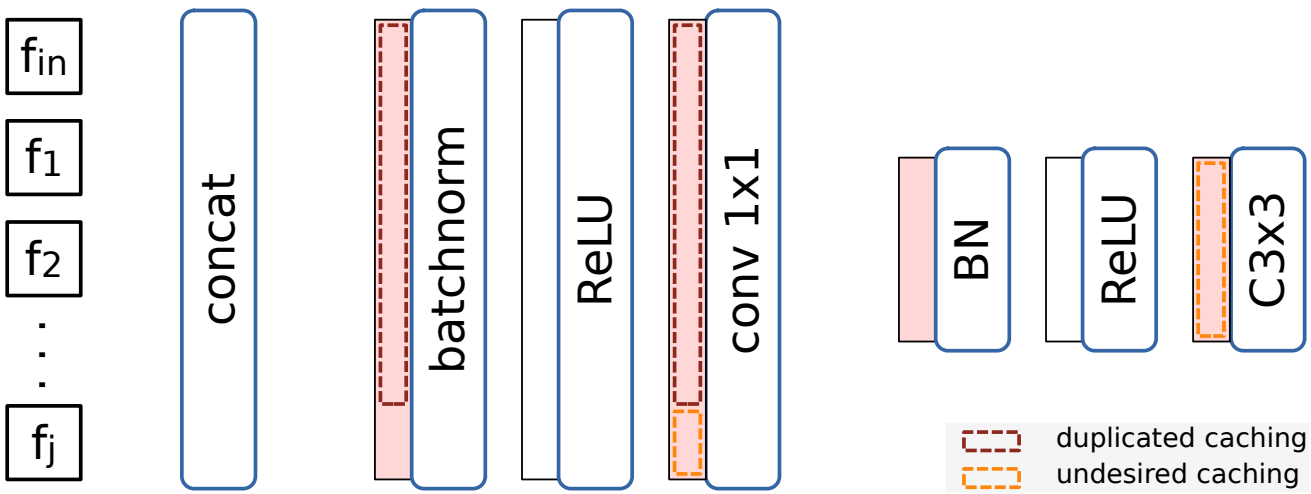
[kreso21phd]

However, DenseNet units are not atomic: let's have a look!

Dense connectivity: footprint - practice

However, DenseNet units are not atomic:

- they consist of a sequence: BN-ReLu-c1x1 BN-ReLU-c3x3
- autograd caches pink tensors; it is unable to notice that it could cache f_is in O(n) instead of their concatenations in O(n^2)
- thus, the default DenseNet caches each unit multiple times (red):

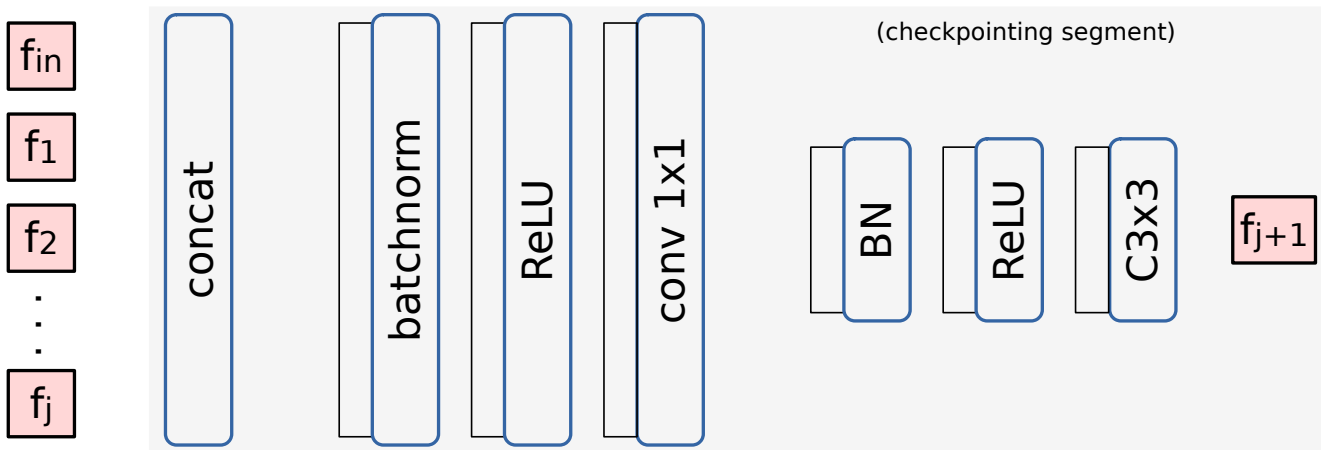


Hence, a popular vote describes DenseNets as memory hungry.

Dense connectivity: checkpointing

Nevertheless, autograd can be instructed to consider the whole convolutional unit as a single node of the computational graph.

The technique is called checkpointing. As a result, only f_i are cached:



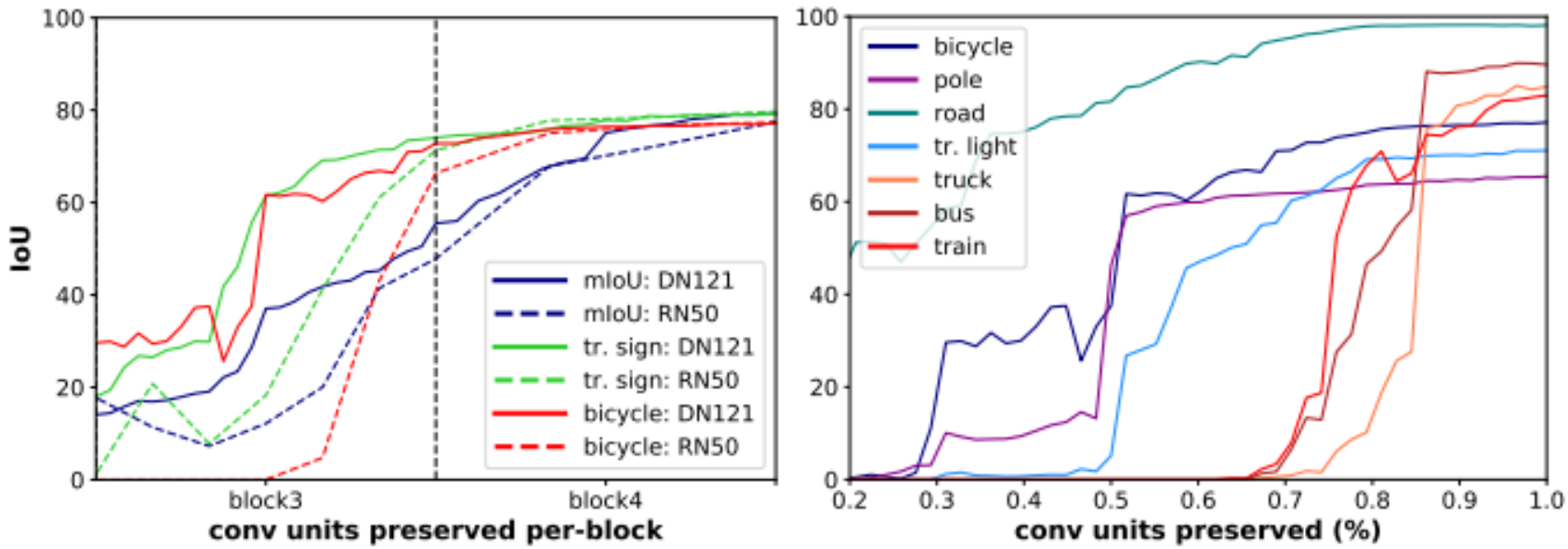
Checkpointing LDN-161: 6-fold memory reduction, 27% more time

This is how we succeeded to train very competitive models on commodity hardware and to deliver competitive research.

Dense connectivity: intuition

We check whether DenseNets really operate by performing useful work in early stages.

We compare ResNet-50 with DenseNet-121 for resilience to layer deletion (without fine-tuning)

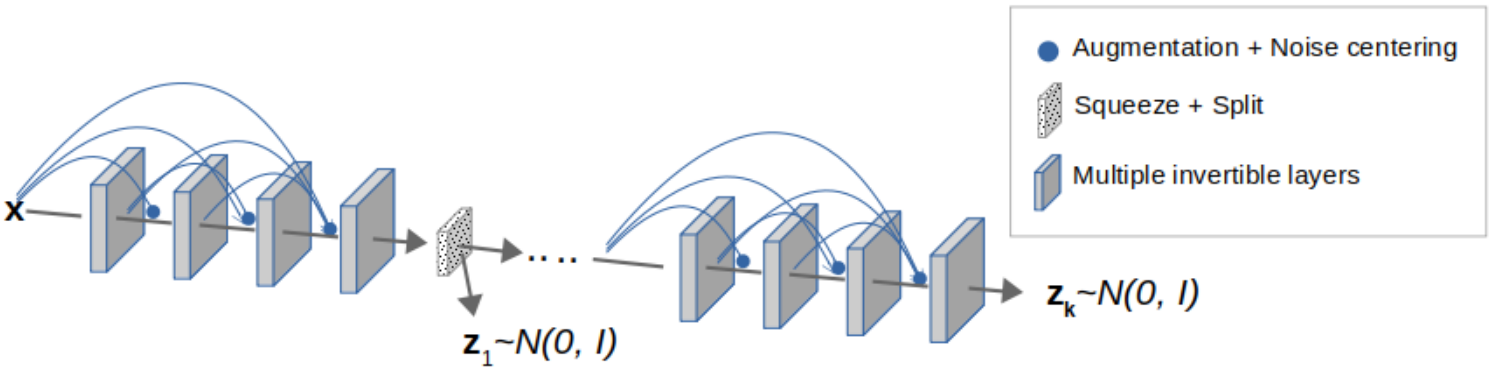


[kreso20tits]

DenseNet-121 wins the challenge; this supports our hypothesis

Dense connectivity: nflows

DenseFlow applies dense connectivity to normalizing flows [grcic21neurips]:



[grcic21neurips]

DenseFlow outperforms approaches trained with 24x more GPU power.

Dataset	Model	GPU type	GPUs	Duration (h)	Likelihood (bpd)
ImageNet32	VFlow [24]	Tesla V100	16	∼1440	3.83
	NVAE [51]	Tesla V100	24	70	3.92
	DenseFlow-74-10	Tesla V100	1	310	3.63

[grcic21neurips]

Inductive bias towards efficient computation of simple features can help in unsupervised learning as well.

Dense connectivity: performance

Cityscapes experiments reveal:

- competitive performance wrt state of the art with much less computations (left, full resolution)
- dilated models (LDDN-121*) underperform wrt ladder models (LDN-121*, half resolution, right)

Method	Backbone	IoU Val	IoU Test	Tflop@1Mpx single scale
LKM [25]	rn50 d32↓	77.4	76.9	0.110†
TuSimple [46]	rn101 d8↓	76.4	77.6	0.720†
SAC-multiple [47]	rn101 d8↓	78.7	78.1	0.720†
ResNet-38 [48]	wrn38 d8↓	77.9	78.4	2.110†
PSPNet [17]	rn101 d8↓	n/a	78.4	0.720†
Multi Task [49]	rn101 d8↓	n/a	78.5	0.720
TKCN [50]	rn101 d8↓	n/a	79.5	0.720†
DFN [51]	rn101 d32↓	n/a	79.3	0.450†
Mapillary [20]	wrn38 d8↓	78.3	n/a	2.110†
DeepLab v3 [19]	rn101 d8↓	79.3	n/a	0.720†
DeepLabv3+ [33]	x-65 d8↓	79.1	n/a	0.710
DRN [52]	wrn38 d8↓	79.7	79.9	2.110†
DenseASPP [21]	dn161 d8↓	78.9	**80.6**	0.500†
LDN121 64→4	dn121 64↓	**80.3**	**80.0**	**0.066**
LDN161 64→4	dn161 64↓	**80.7**	**80.6**	0.139

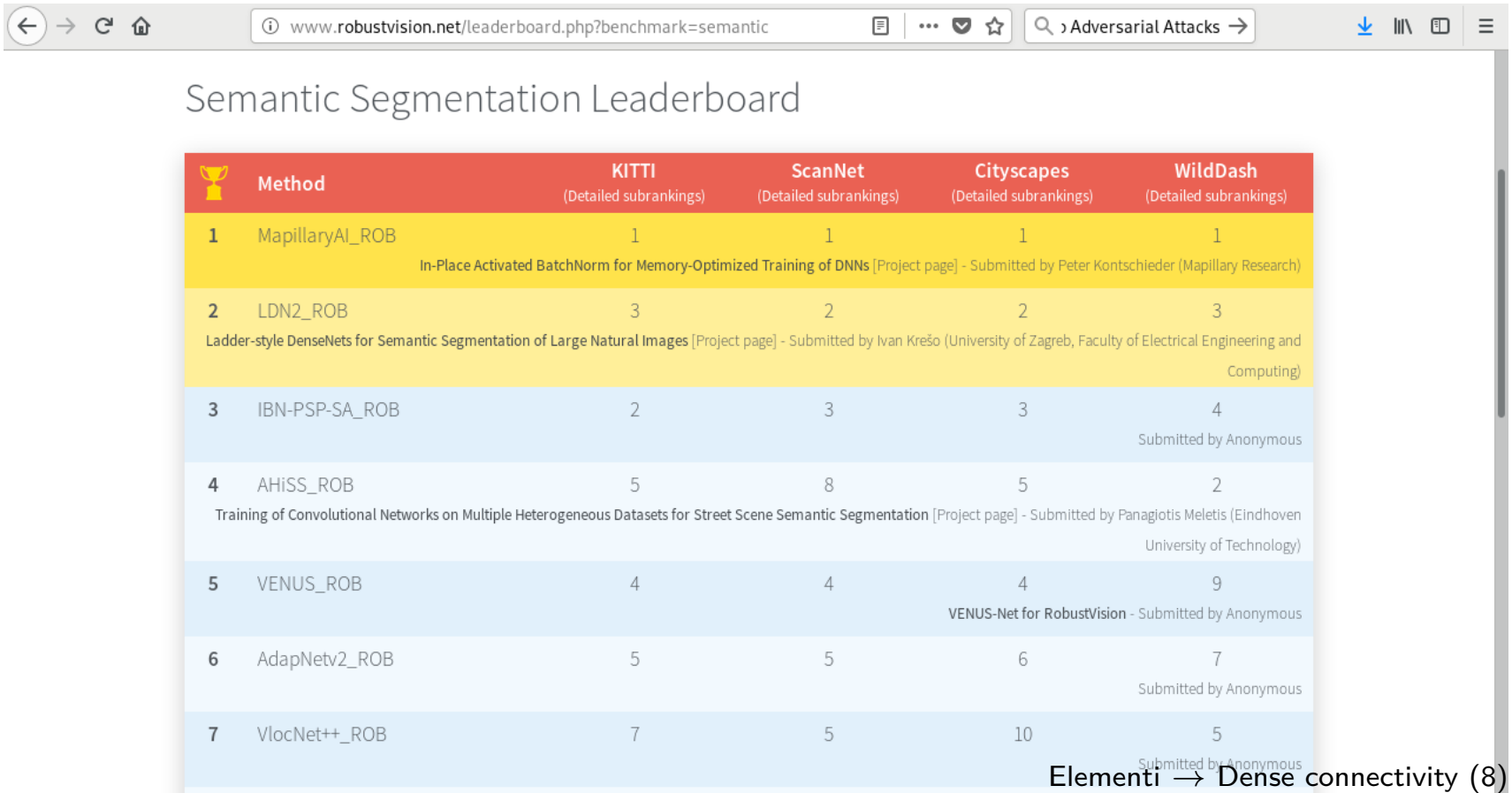
Method	Class IoU	Class iIoU	Cat. IoU	Model size	FLOP 1MPx
DN121 32↓	66.2	46.7	78.3	8.2M	56.1G
LDN121 64→4	75.3	54.8	88.1	9.5M	66.5G
LDN121 32→4	**76.6**	57.5	88.6	9.0M	75.4G
LDN169 32→4	75.8	55.5	88.4	15.6M	88.8G
LDN121 32→2	**77.5**	**58.9**	89.3	9.4M	154.5G
ResNet18 32→4	70.9	49.7	86.7	13.3M	55.7G
ResNet101 32→4	73.7	54.3	87.8	45.9M	186.7G
ResNet50 32→4	73.9	54.2	87.8	26.9M	109.0G
DPN68 32→4	74.0	53.0	87.8	13.7M	59.0G
DDN-121 8↓	72.5	52.5	85.5	8.2M	147.8G
LDDN-121 8→4	75.5	55.3	88.3	8.6M	174.8G
LDDN-121 16→4	75.8	55.9	88.4	8.9M	87.0G

[kreso21tits]

Dense connectivity: RVC 2018

We have used LDN-169 at Robust Vision Challenge 2018:

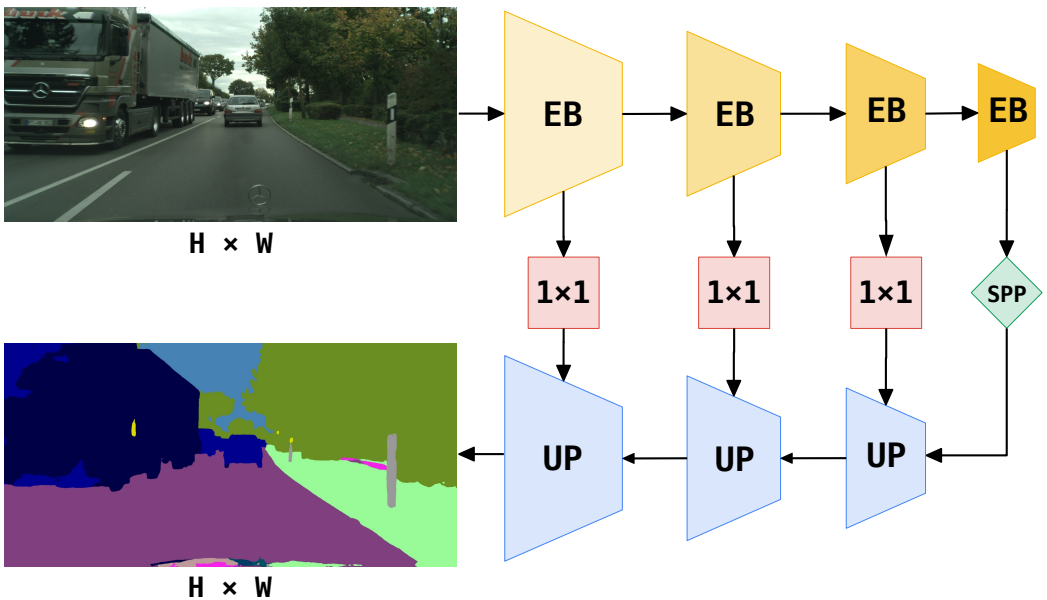
- evaluation of one model on four benchmarks [kreso18arxiv]
- we ranked #2 out of 10 in spite of training on one GPU
- the winners could train on much more data due to having 8xV100

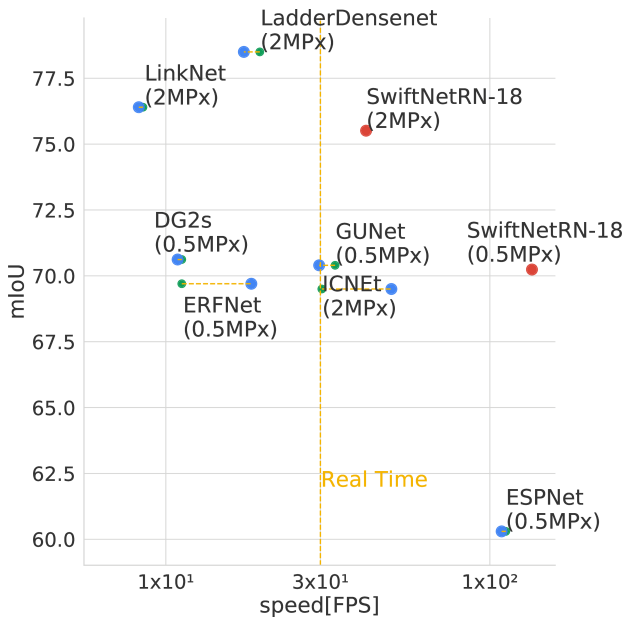


Efficient inference: concept

SwiftNet --- efficient variant of Ladder-DenseNet based on ResNet-18:

- very fast training and inference
- outperformed prior real-time models by a large margin
- still a competitive baseline for low-power applications



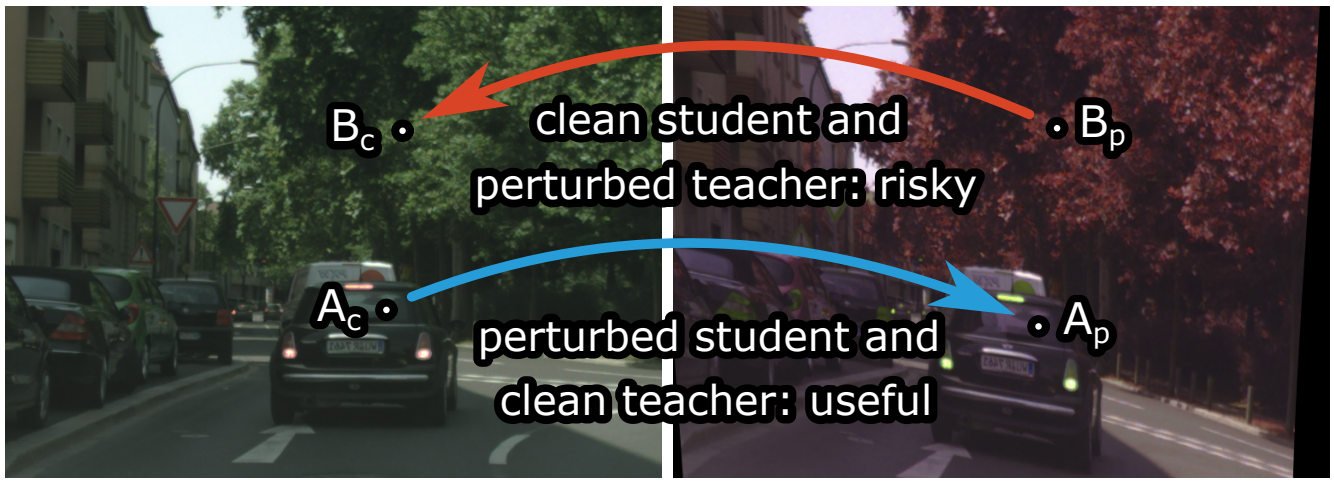


[orsic19cvpr]

Efficient inference: semi-supervised

Semi-supervised learning uses labeled and unlabeled data:

- extremely important since it relaxes dependence on labeled data
- our work sheds additional light on widely used consistency loss (and proposes a state-of-the-art perturbation model)



[grubisic21mva]

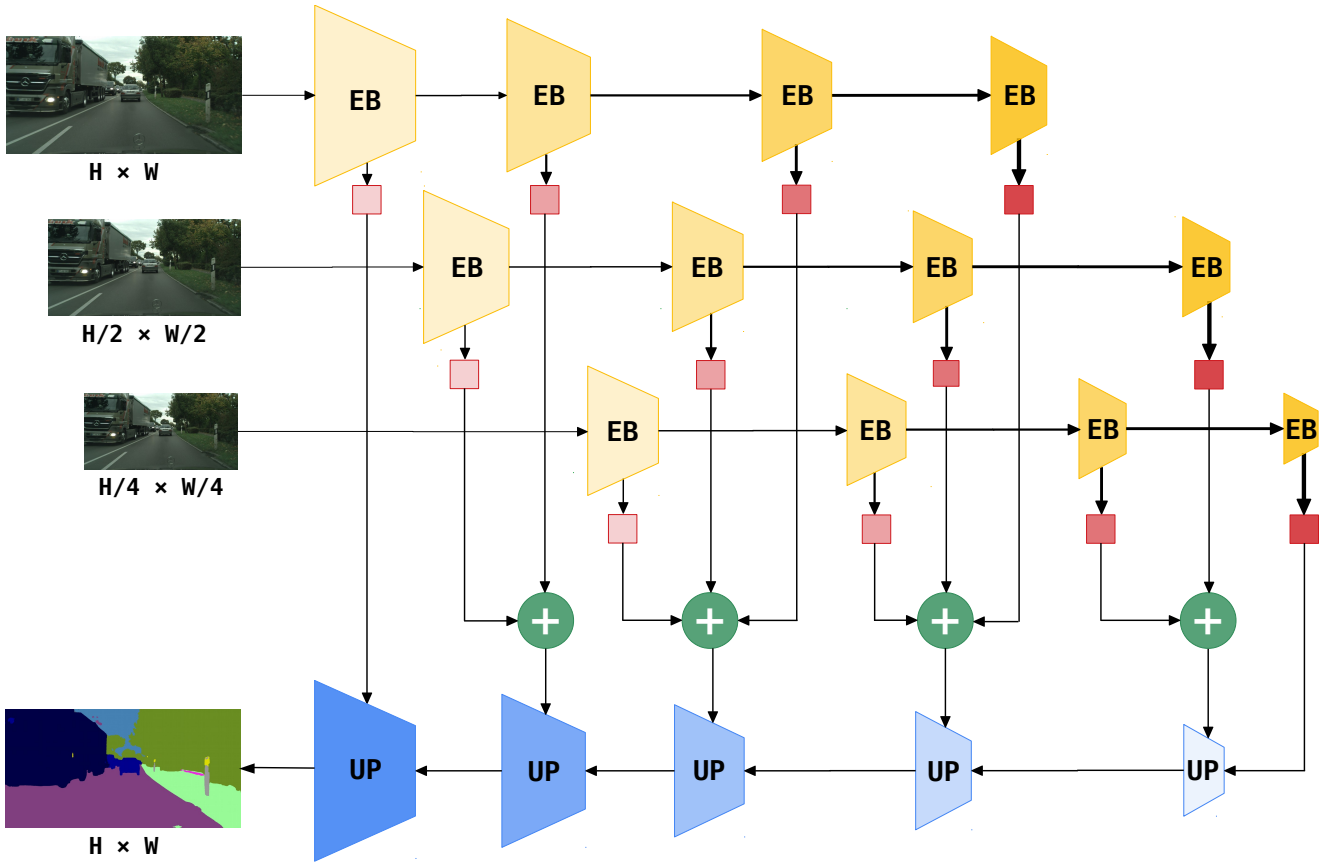
In comparison to widely used DeepLabV2-RN101:

- SwiftNet-RN18 delivers comparable performance
- SwiftNet-RN18 requires 12x less memory and 12x faster inference.

Efficient inference: pyramidal fusion

Scale-equivariant recognition and cross-scale upsampling:

- decreases the speed for only 30% due to strong subsampling
- a strong contender both in embedded and large-capacity setups
- confirms utility of inductive bias on our datasets.



Efficient inference: RVC 2020

We have used pyramidal SwiftNet at Robust Vision Challenge 2020:

- submit the same model to 7 benchmarks with incompatible labels
- our strengths: SNPyr, DN161ckpt, NLL+
- A major problem: datasets have incompatible taxonomies
  - pickups labeled as trucks (VIPER), cars (Vistas) and vans (ADE20k)

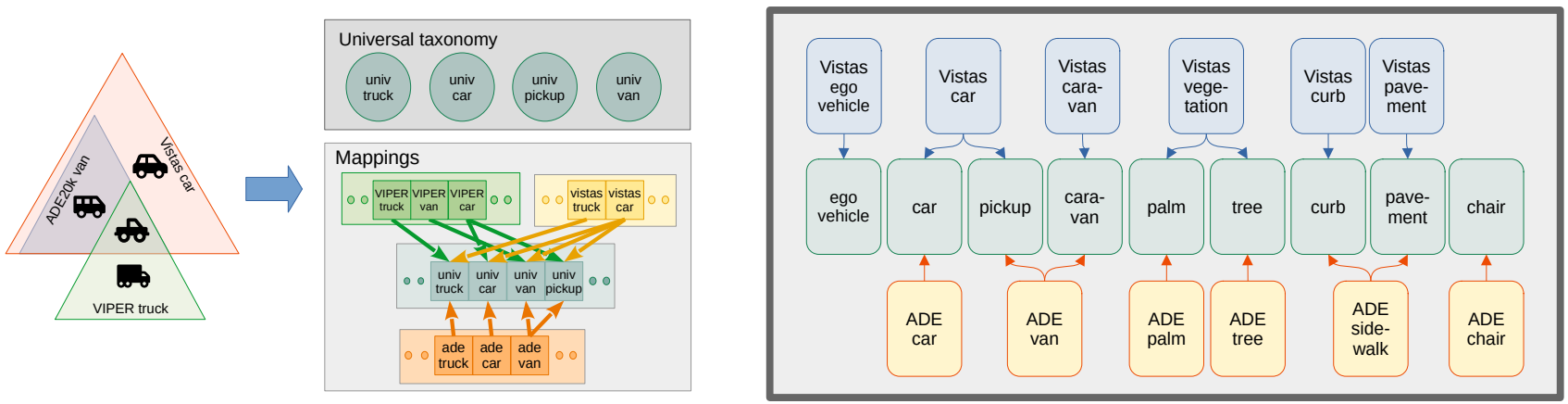
[bevandic22wacv]

Efficient inference: RVC 2020 (2)

We address this problem by devising a flat universal taxonomy:

- dataset classes are expressed as unions of universal classes
- probability of a dataset class y is a sum of probabilities (NLL+) of universal classes u'

$$\mathcal{L}^{\text{NLL+}}(x, y, m_{\mathbb{S}_d}) = -\ln \sum_{u' \in m_{\mathbb{S}_d}(y)} \mathrm{P}(U = u' \mid \boldsymbol{x})$$

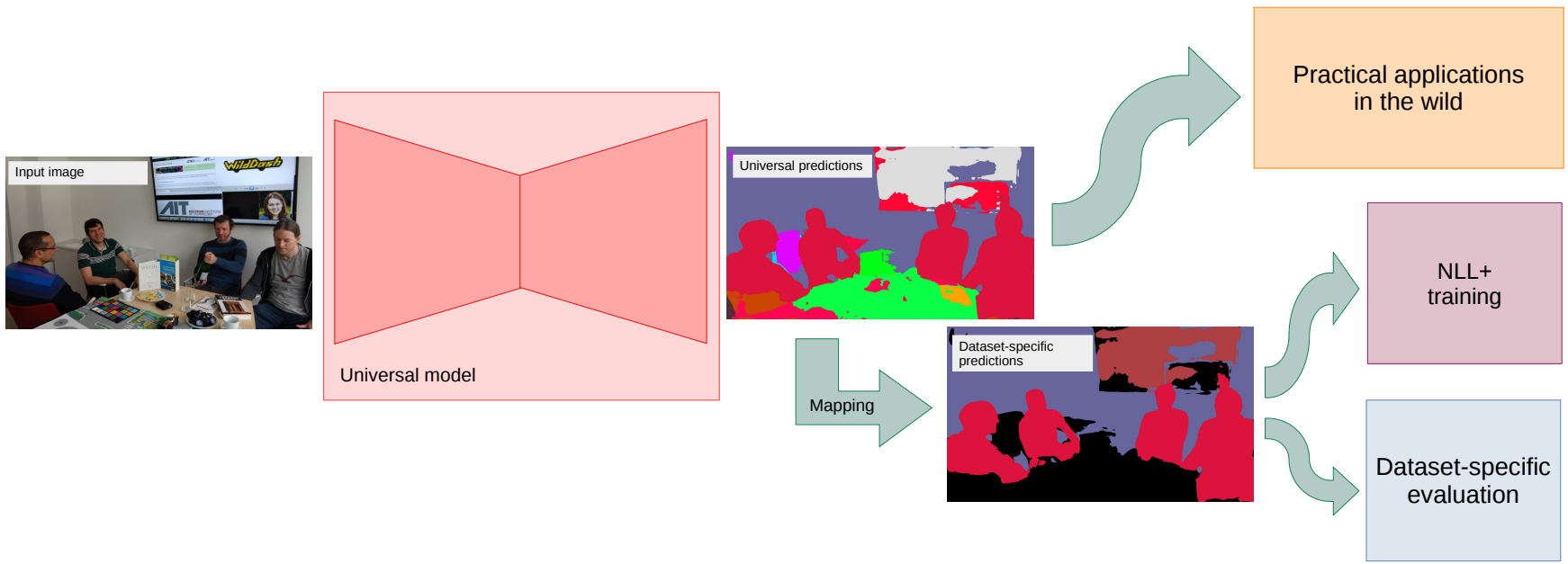


[bevandic22wacv]

Efficient inference: RVC 2020 (3)

Our models infer universal classes that allow:

- interpretable inference in the wild
- training on original datasets (no relabeling required)
- evaluating on original datasets



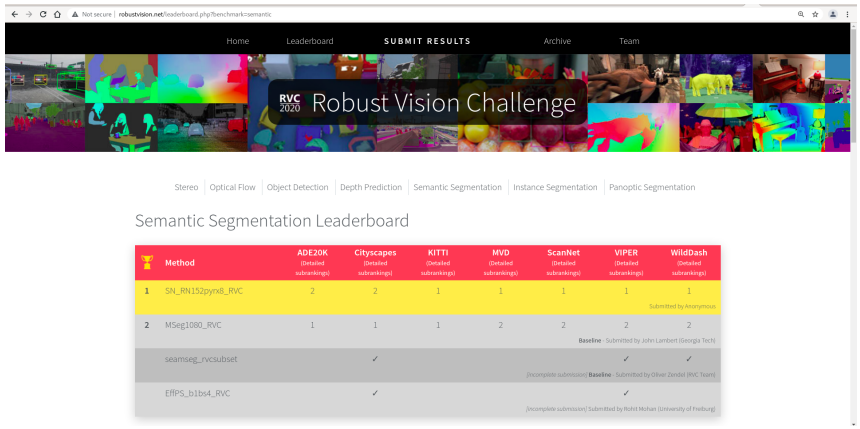
[bevandic22wacv]

.

Petra's paper on automatic taxonomies will hit Arxiv in a day or two

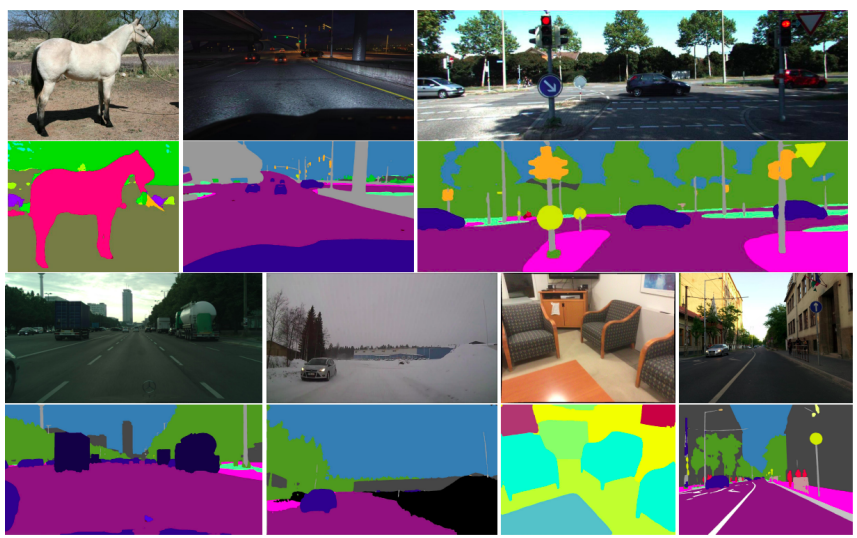
Efficient inference: RVC 2020 (4)

We achieved rank #1 on the semantic segmentation task:



[orsic20arxiv]

The trained model can segment test images from multiple domains:



[bevandic22wacv]

Efficient inference: WildDash 2

Our ROB 2020 model sets the state of the art on the most advanced road driving benchmark:

- hand-picked very hard scenes with various kinds of domain shift

	Meta AVG	Classic				Negative	Impact (IoU class)									
Algorithm	IoU Class	IoU Class	iIoU Class	IoU Cat.	iIoU Cat.	IoU Class	Blur	Coverage	Distortion	Hood	Occ.	Overexp.	Particles	Screen	Underexp.	Var.
SN_DN161_fat_pyrx8	46.8%	51.0%	43.9%	71.4%	65.5%	32.6%	-7%	-11%	-5%	-9%	-3%	-2%	-7%	**-22%**	-8%	-8%
SN_DN161s3pyrx8	45.6%	49.8%	41.6%	71.3%	65.3%	31.0%	-10%	-6%	-6%	-10%	-3%	-3%	-6%	**-20%**	-9%	-10%
SN_RN152pyrx8_RVC	45.4%	48.9%	42.7%	70.1%	64.8%	32.5%	-6%	-7%	-5%	-7%	-1%	-2%	-7%	**-19%**	-11%	-3%
seamseg_rvcsubset	37.9%	41.2%	37.2%	63.1%	58.1%	30.5%	-16%	-17%	0%	-7%	-4%	-14%	-18%	**-31%**	-14%	-7%
Tong	37.2%	41.0%	41.2%	65.2%	53.5%	26.0%	-18%	-9%	-5%	-16%	-2%	-13%	-12%	**-24%**	-10%	-1%
seamseg_mvd_ss	37.1%	41.3%	36.9%	63.4%	55.7%	26.6%	-15%	-14%	0%	-11%	-4%	-11%	-30%	**-36%**	-20%	-10%
SIW	36.5%	41.0%	38.6%	65.8%	53.1%	24.1%	-16%	-17%	-6%	-14%	-2%	-7%	-19%	**-23%**	-10%	-6%
hs1	35.7%	40.0%	38.0%	64.8%	52.3%	23.0%	-17%	-10%	-8%	-18%	-1%	-15%	-11%	**-27%**	-9%	-9%
MSeg1080_RVC	35.2%	38.7%	35.4%	65.1%	50.7%	24.7%	-15%	-11%	-9%	-19%	-3%	-14%	-6%	**-25%**	-8%	-13%
hs	34.4%	38.4%	36.2%	64.2%	52.1%	22.3%	-19%	-11%	-8%	-18%	0%	-13%	-15%	**-29%**	-11%	-6%
EffPS_b1bs4sem_RVC	32.2%	35.7%	24.4%	63.8%	56.0%	20.4%	-10%	-6%	-4%	-7%	-1%	-7%	-10%	**-25%**	-8%	-6%

[bevandic22wacv]

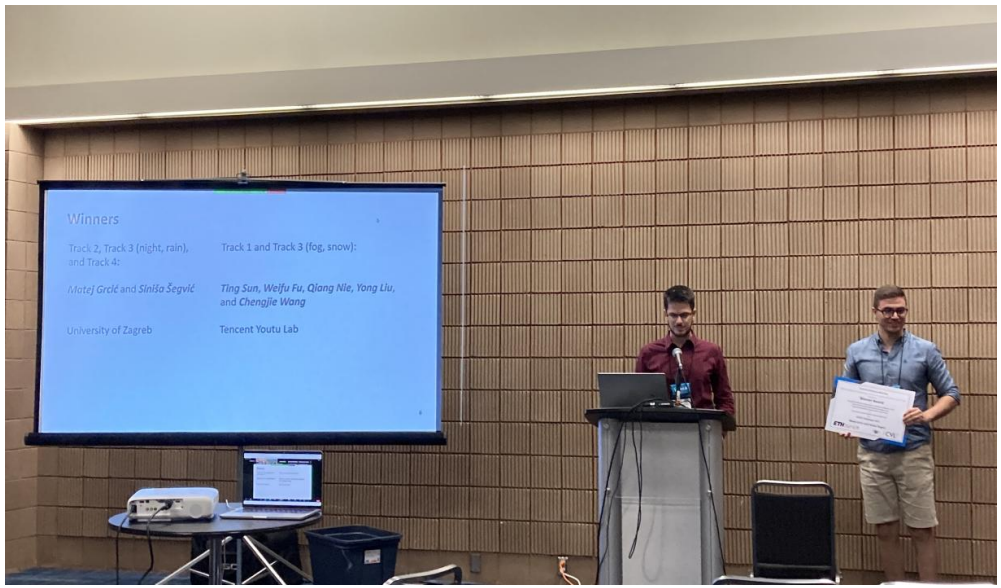
Efficient inference: WildDash 2 (2)

[bevandic22wacv]

Efficient inference: ACDC 2022

Matej had a fruitful participation at the ACDC challenge:

- Adverse conditions: snow, fog, rain, night
- Model: pyramidal SwiftNet with ConvNext-Large
- Semi-supervised setup: 40K labeled + 250k unlabeled images
- Matej won in 2.5 out of 4 tracks

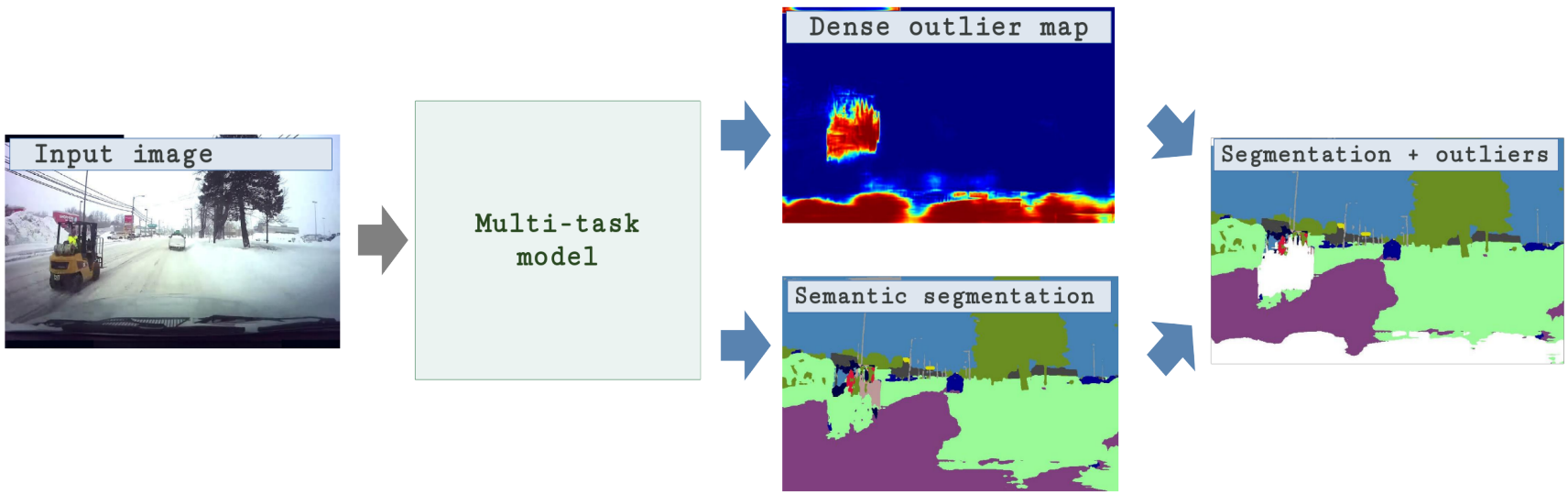


[grcic22v4as]

Open-set: intro

Deep models underperform on outliers:

- especially problematic in dense prediction where only part of the scene may be anomalous
- several recent datasets address that problem, eg. StreetHazards, Fishyscapes, Segment Me If You Can
- the problem can be addressed with open-set recognition models.

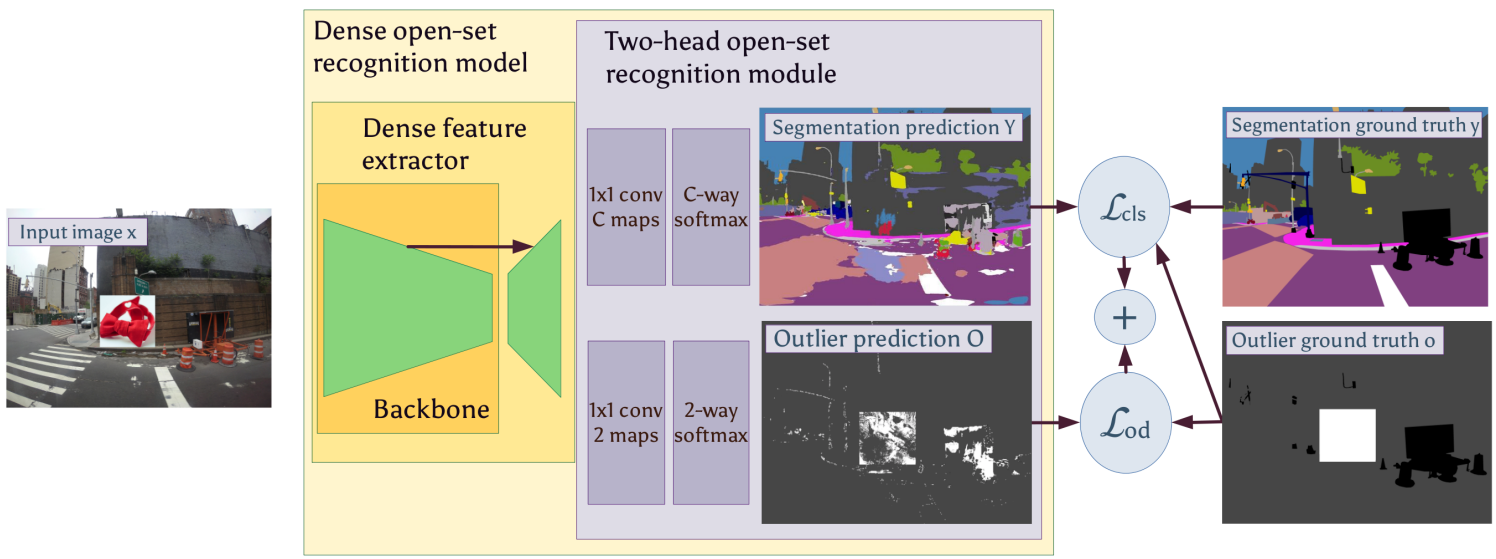


[bevandic19gcpr]

Open-set: noisy negatives

Open-set models can be obtained by training on noisy negative data [bevandic19gcpr]

Interestingly, we need to train on mixed content images in order to be able detect outlier objects in inlier context



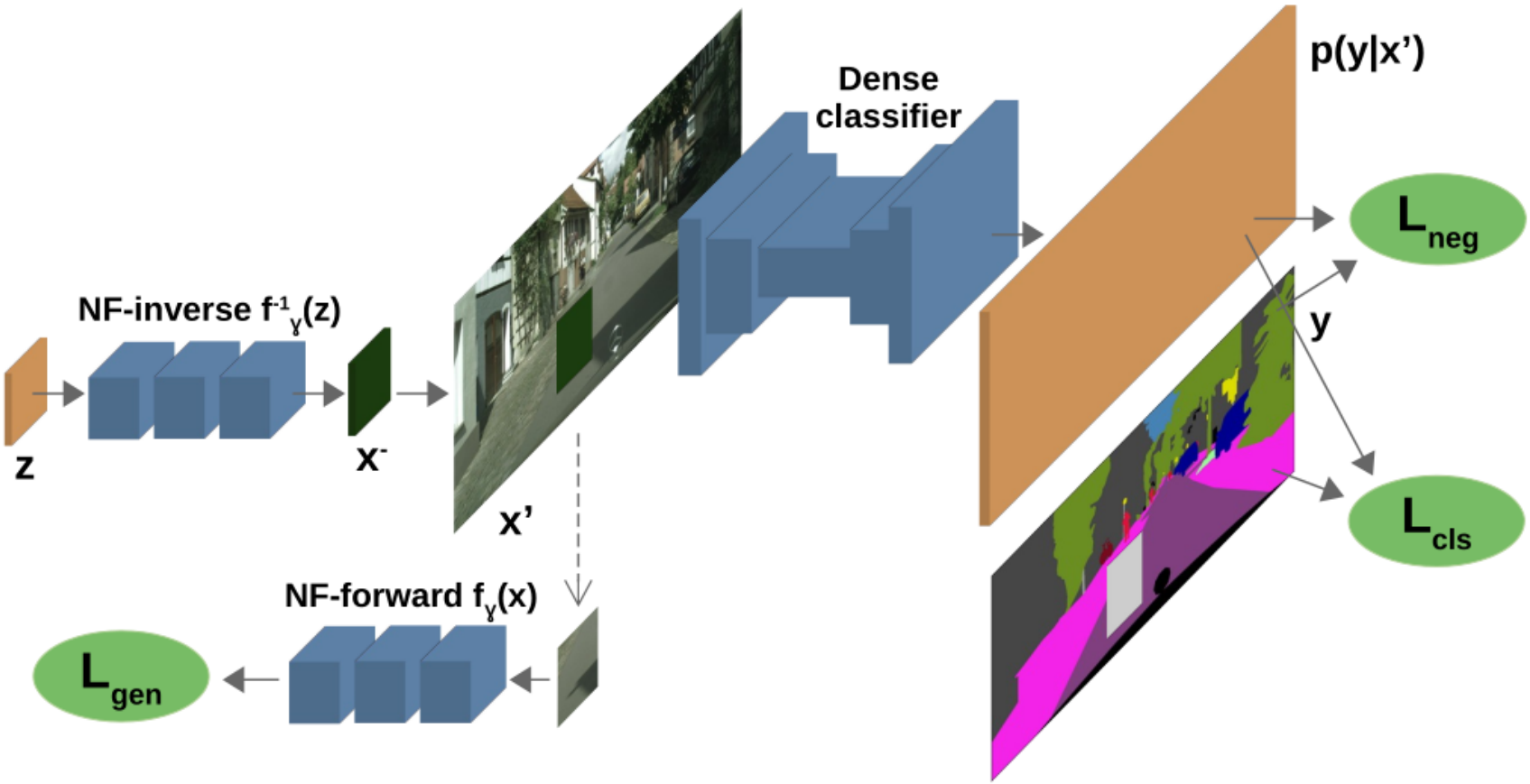
[bevandic22ivc]

This suggests that deep models are lazy: what you get is what you ask.

Open-set: synthetic negatives

Training with auxiliary negatives effective but perhaps over-optimistic

Use surrogate negatives which we generate by jointly optimizing inlier likelihood and uniform posterior in synthetic samples [lee18iclr]:



[grcic21visapp]

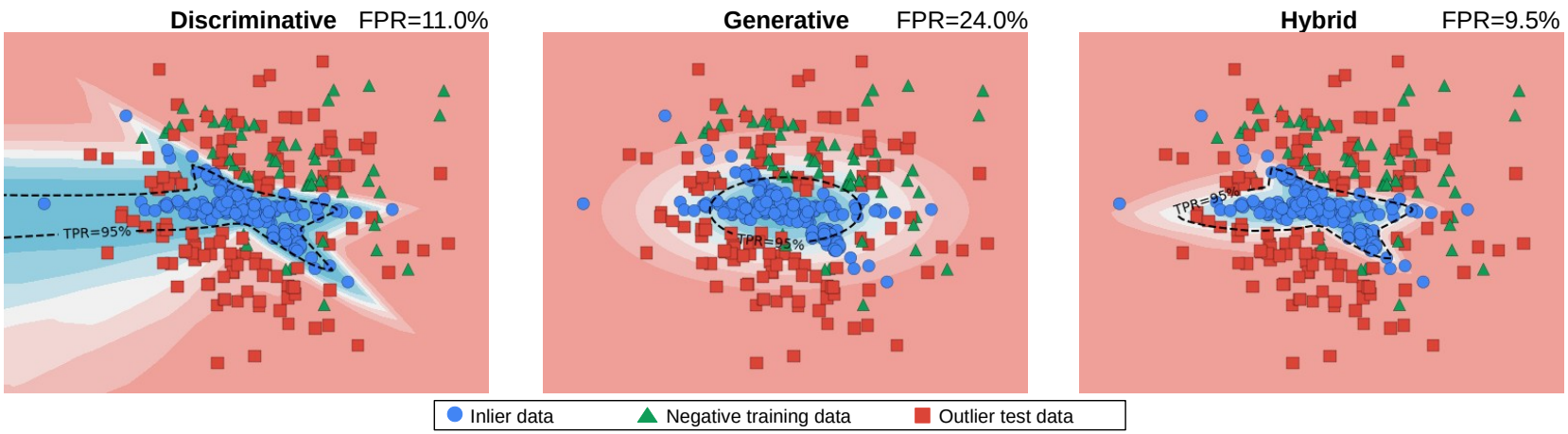
Open-set: hybrid modeling

Interpret the classifier as an energy-based model: [grathwohl20iclr]

$$p(Y, \mathrm{x}) = \frac{\exp s}{Z}$$

$$p(\mathrm{x}) = \frac{\sum_j \exp s_j}{Z}$$

Train semantic segmentation by encouraging low p(x) in negatives:



[grcic22eccv]

Challenges and Opportunities: two perspectives

Optimistic perspective:

- unprecedented progress, healthy rate of improvement
- we would be busy sorting details for at least a decade even with no further development (unlikely)
- Moore law still applies: the computing power will increase

Pessimistic perspective:

- tough competition, many smart people produce at full speed
- unreasonable to expect faster rate of improvement
- hardware improves slowly and wastes a lot of energy;
- bitter lessons suggest that substantial advances will require a lot of computing power.

Clearly, there is some uncertainty ahead.

Challenges and Opportunities: a view

Thomas Edison said in 1895:

> *It is apparent to me that the possibilities of the aeroplane, which two or three years ago were thought to hold the solution to the [flying machine] problem, have been exhausted, and that we must turn elsewhere.* [https://www.xaprb.com/blog/flight-is-impossible/]

Challenges and Opportunities: a view (2)

In spite of abundant scepticism, the Wright brothers flew in 1903:

[wikipedia]

Wilbur Wright delivered the following speech in 1908:

> *I know of only one bird, the parrot, that talks, and he can't fly very high.*

Challenges and Opportunities: a view (3)

A honorary mention goes to Otto Lilienthal who flew in 1894 (though without motors):

[wikipedia]

Challenges and Opportunities: future work

- Learning with incomplete supervision:
  - discriminative vs generative vs self-supervised
  - huge industrial value due to relaxed dependence on labels
- Deep learning for reconstruction:
  - adapt classic approaches for end-to-end learning
- Transformers
  - they may offer a way to smarter vision
- Increasing robustness to distribution shifts
  - multi-domain, outliers, adversarial examples, cross-dataset learning
- New kinds of inductive bias
  - limited by imagination
- New hardware
  - Tesla NPU: 37 TOPS, 36W
  - Google TPUv3: 100 TFLOPS, 450W

Conclusions : Takeaway

Useful pieces of inductive bias for natural image understanding:

- translational equivariance (convolutional layers)
- scale equivariance (image pyramids, shared parameters)
- recognition is harder than finding borders when semantics is known (ladder-style upsampling)
- some useful features are more easily recognized than others (DenseNets)
- allowing the model to see a wider context (pyramidal pooling, multi-resolution processing)
- existence of anomalies (negative training data)
- existence of anomalous objects (mixed-content images)

Thank you for your attention!

Questions?

This presentation would not have been possible without insightful ideas and hard work of Ivan Krešo, Marin Oršić, Petra Bevandić, Josip Šarić, Ivan Grubišić, Matej Grcić, Marin Kačan and Iva Sović.

This research has been supported by Croatian Science Foundation (MULTICLOD, ADEPT), ERDF (DATACROSS, A-UNIT, SAFETRAM), Rimac automobili, Microblink, Gideon brothers, Romb technologies, Končar, UniZg-FPZ, and VSITE.