

Elements of Learning Algorithms for Natural Scene Understanding

constructing software by expressing bias and loss

Siniša Šegvić
UniZg-FER D307

AGENDA

- Introduction: building software that learns from data
- Part 1: Elements of deep models for natural scene understanding
- Part 2: Overview of our industrial collaboration
- Part 3: Challenges and Opportunities
- Conclusion: prospects, space for improvement

INTRODUCTION : CHANGE OF TIDE

Deep learning caused profound changes into computer vision methodology

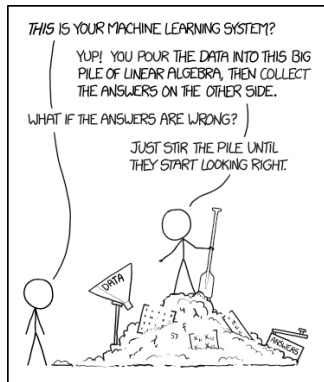
Many of our beloved methods rapidly rapidly fell out of luck, eg:

- 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.

Deep learning can automate only implementation details (eg. feature extraction), the brains of our methods still have to be designed by us:

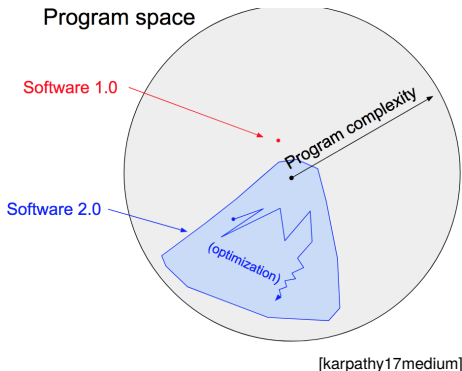
- preference towards some solutions (aka inductive bias)
[kreso21tits,grcic21neurips,orsic21pr,saric20cvpr]
- loss formulation to exploit incompatible labels [bevandic22wacv]
- learning algorithms for exploiting unlabeled data [grubisic21mva]

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

INTRODUCTION : RED OR BLUE? (2)

In general, our today's job consists in:

- outlining a fairly large class of solutions (blue region)
- letting the optimization arrange details according to the data



Much more powerful than classic software development (red)!

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:

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

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]

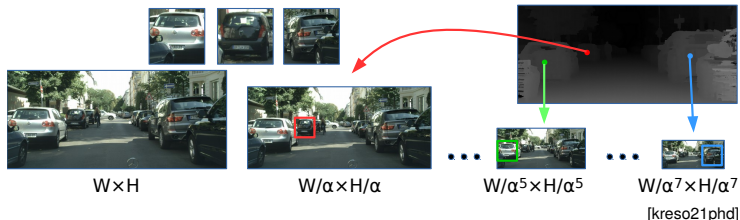
SCALE INVARIANCE : HAND-CRAFTED PERSPECTIVE

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

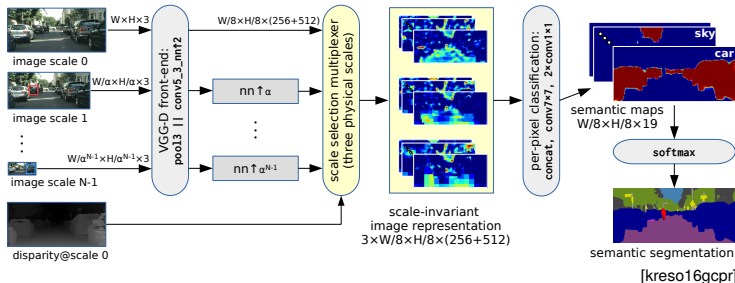


SCALE INVARIANCE : HAND-CRAFTED PERSPECTIVE (2)

Efficient (GPU friendly) implementation:

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

This simplifies things to the model by presenting all parts of the scene as if they were filmed from three canonical distances.



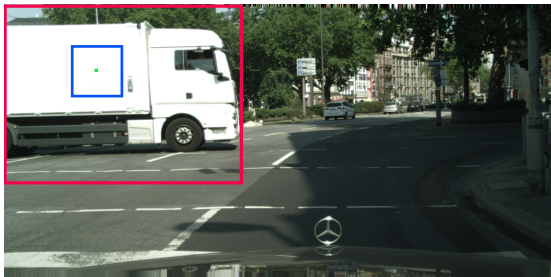
SCALE INVARIANCE : HAND-CRAFTED PERSPECTIVE (3)

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:

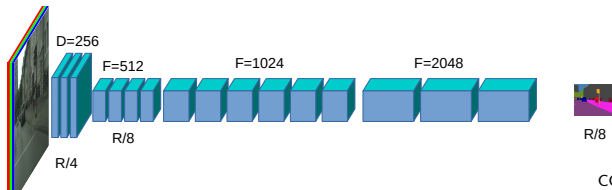
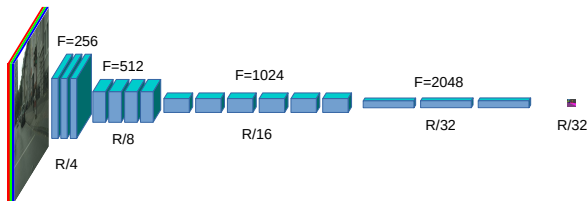


LADDER DENSENET: DILATION?

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 DENSENET: BRUTE FORCE OR EFFICIENT?

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

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

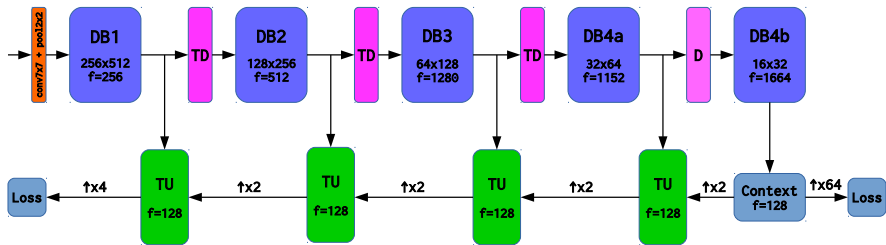
Instead, we chose to compete by making our models more efficient:

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

LADDER DENSENET: CONCEPT

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

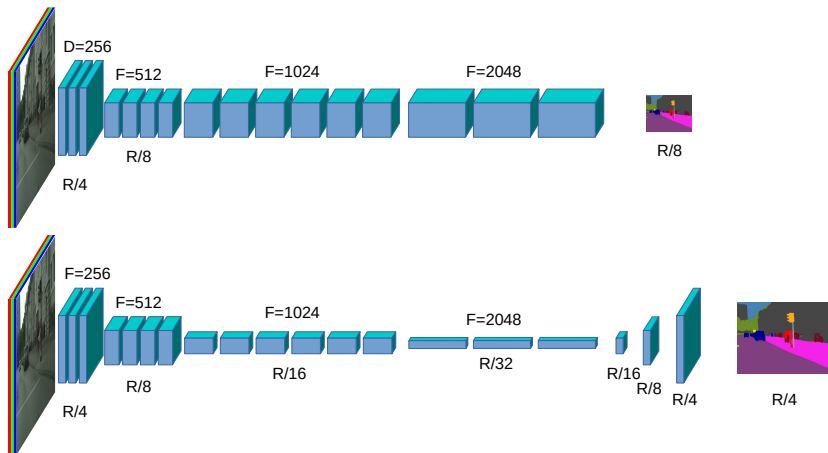
- most layers and most capacity assigned to the backbone
- context recognition module increases the receptive field of the most compressed representation
- ladder-style upsampling blends low-resolution semantics with high-resolution details



LADDER DENSENET: BIAS

Inductive bias of ladder-style upsampling:

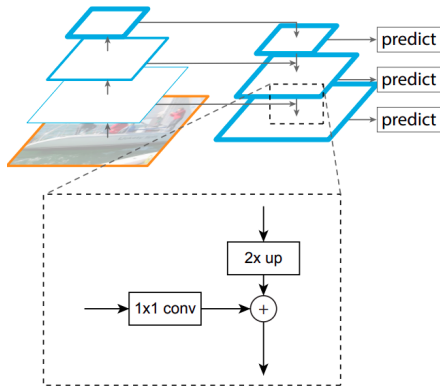
- recognition receives more capacity than border refinement
- bonus: much less computations (blue bricks) than dilated models



LADDER DENSENET: RELATED WORK

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

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

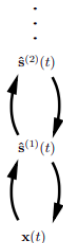


LADDER DENSENET: RELATED WORK (2)

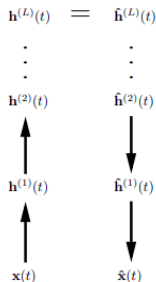
Ladder-style reconstruction has been invented for autoencoders:

- deep layers focus on abstract invariant features
- shallow layers keep information to reconstruct details
- skips improve gradient exposure and speed-up the learning

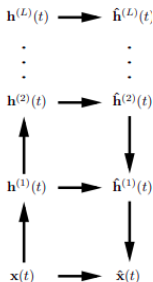
Hierarchical
latent variable
model



Standard
autoencoder
network



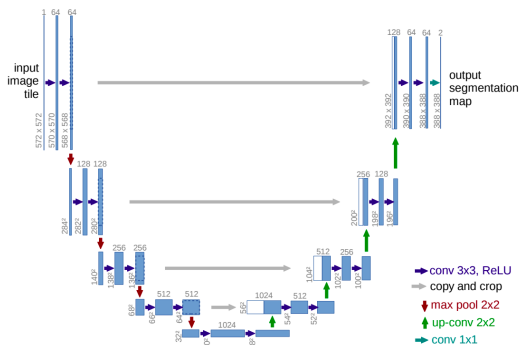
Ladder
autoencoder
network



[valpola14arxiv]

LADDER DENSENET: RELATED WORK (3)

Ladder-style upsampling has also been used in the UNet architecture:



[ronneberger15miccai]

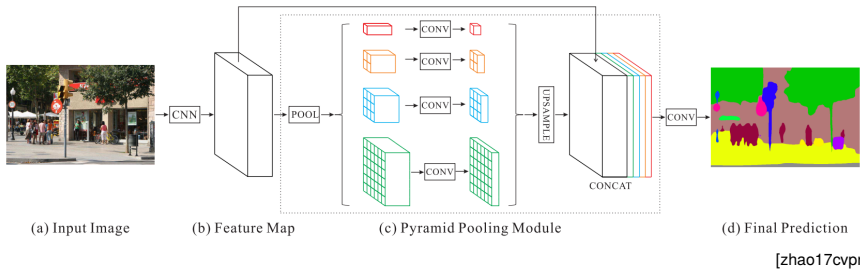
Our architecture outperforms UNets due to following advantages:

- asymmetric design: +generalization, -computations
- increased receptive field due to context/pyramid pooling
- standard recognition backbone allows pre-training

LADDER DENSENET: PYRAMID POOLING

Convolutional pyramid pooling [zhao17cvpr]:

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

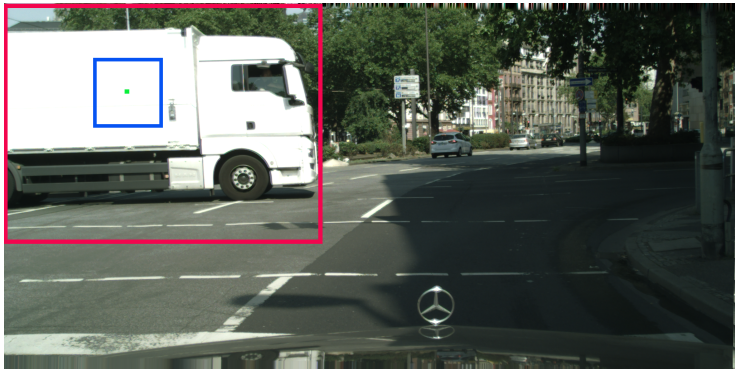


Previous uses of pyramid pooling:

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

LADDER DENSENET: PYRAMID POOLING (2)

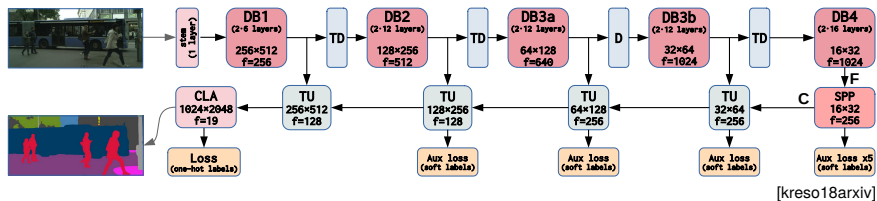
Pyramid pooling allows the model to recognize pixels on smooth surfaces by relying on context:



[kreso16gcp]

LADDER DENSENET: PYRAMID POOLING (3)

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



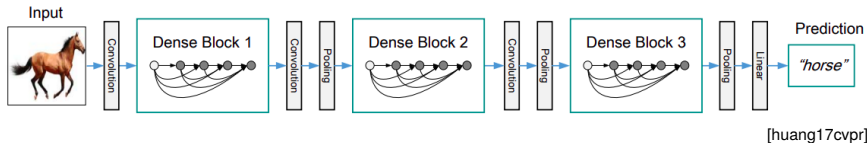
This achieves similar (or better) effects with much less computational power [kreso21tits].

LADDER DENSENET: BACKBONE

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

A DenseNet model consists of 3-5 processing blocks:

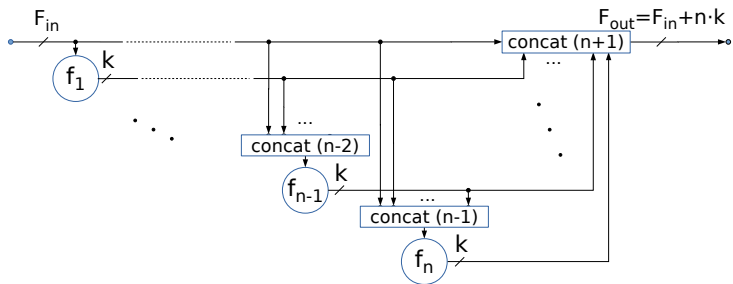
- multi-unit convolutional modules (6-100+ convolutions)
- all these convolutions operate at the same resolution
- other architectures (AlexNet, VGG, ResNet) have similar structure



LADDER DENSENET: BACKBONE (2)

A DenseNet block relies on dense connectivity and concatenations:

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



[kreso21phd]

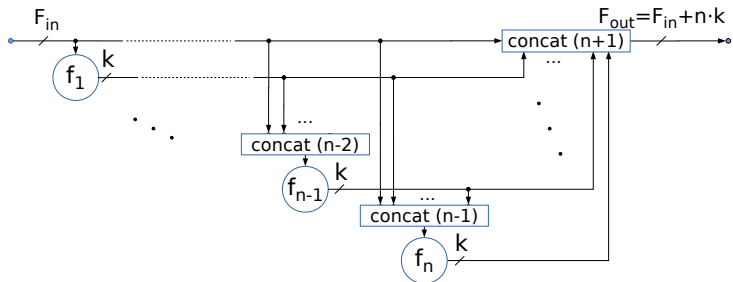
The solution is expressed in terms of features with differing capacities

- this inductive bias makes DenseNets very efficient

LADDER DENSENET: BACKBONE (3)

DenseNets also have a great potential to reduce the memory footprint:

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



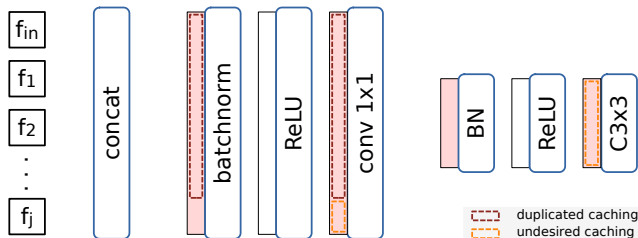
[kreso21phd]

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

LADDER DENSENET: BACKBONE (4)

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_i s in $O(n)$ instead of their concatenations in $O(n^2)$
- thus, the default DenseNet caches each unit multiple times (red):

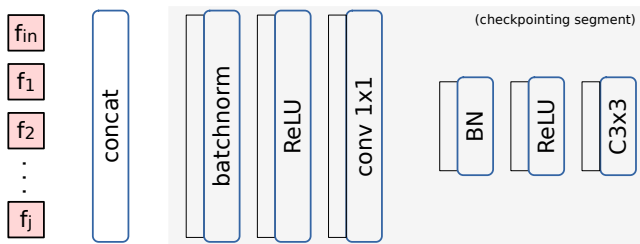


Hence, a popular vote describes DenseNets as memory hungry.

LADDER DENSENET: BACKBONE (5)

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:



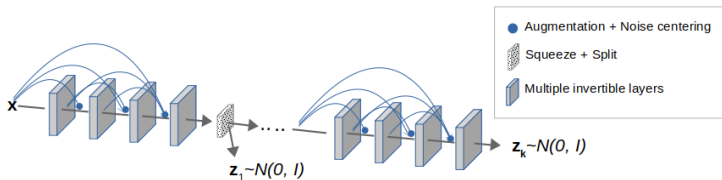
Checkpointing LDN-161: 6-fold memory reduction, 27% more time

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

LADDER DENSENET: BACKBONE (6)

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.

LADDER DENSENET: EXPERIMENTS

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		Tflop@1Mpx single scale
		Val	Test	
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		Cat. IoU	Model size	FLOP 1MPx
	IoU	iloU			
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]

LADDER DENSENET: RVC 2018

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

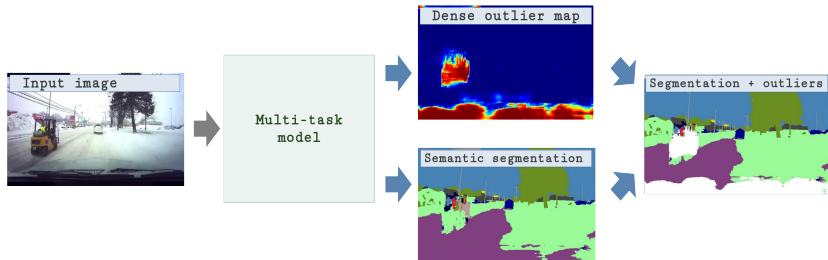
- evaluation of one model on four benchmarks [kreso18arxiv]
- we ranked #2 out of 10 in spite of training on one GPU
- the winners could train on much more data due to having 8xV100

Method	KITTI (Detailed subrankings)	ScanNet (Detailed subrankings)	Cityscapes (Detailed subrankings)	WildDash (Detailed subrankings)
1 MapillaryAI_ROB In-Place Activated BatchNorm for Memory-Optimized Training of DNNs [Project page] - Submitted by Peter Kortschieder (Mapillary Research)	1	1	1	1
2 LDN2_ROB Ladder-style DenseNets for Semantic Segmentation of Large Natural Images [Project page] - Submitted by Ivan Kreso (University of Zagreb, Faculty of Electrical Engineering and Computing)	3	2	2	3
3 IBN-PSP-SA_ROB	2	3	3	4 Submitted by Anonymous
4 AHISS_ROB Training of Convolutional Networks on Multiple Heterogeneous Datasets for Street Scene Semantic Segmentation [Project page] - Submitted by Panagiotis Meletis (Eindhoven University of Technology)	5	8	5	2
5 VENUS_ROB	4	4	4	9 VENUS-Net for RobustVision - Submitted by Anonymous
6 AdapNetv2_ROB	5	5	6	7 Submitted by Anonymous
7 VlocNet++_ROB	7	5	10	5 Submitted by Anonymous

LADDER DENSENET: OPEN-SET

Deep models underperform on outliers:

- several recent datasets address that problem, eg. StreetHazards, Fishyscapes, Segment Me If You Can
- the problem can be addressed with open-set recognition models.

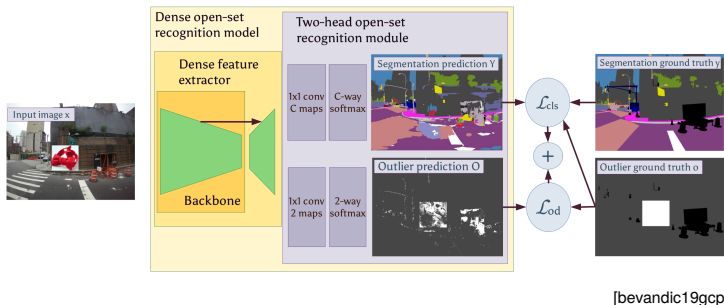


[bevandic19gcpr]

LADDER DENSENET: OUTLIERS (2)

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

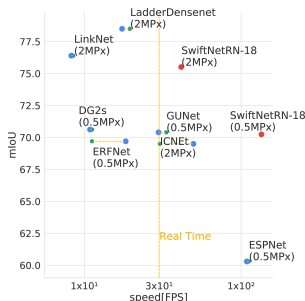
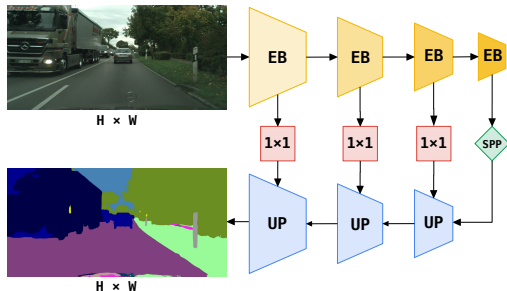


This suggests that deep models are lazy: what you get is what you ask.

SWIFNet: 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

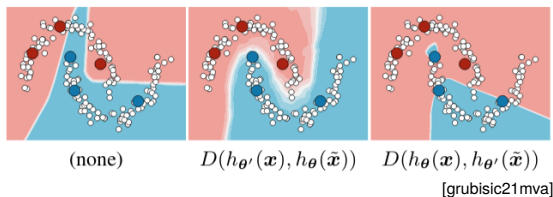


[orsic19cvpr]

SWIFTNET: 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)



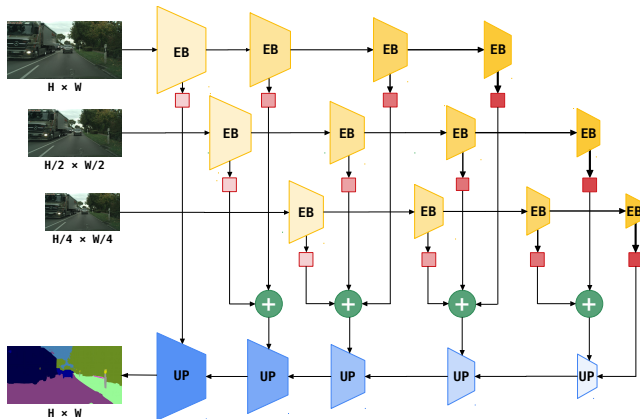
In comparison to widely used DeepLabV2-RN101:

- SwiftNet-RN18 delivers comparable performance in fully-supervised and semi-supervised setups
- SwiftNet-RN18 requires 12x less memory and 12x faster inference.

SWIFTNET: PYRAMIDAL SWIFTNET

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.



SWIFTNET: RVC 2020

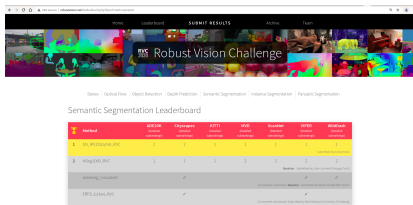
We have used pyramidal SwiftNet at Robust Vision Challenge 2020:

- submit the same model to 7 benchmarks with incompatible labels
- our strengths: SNPyr, DN161ckpt, NLL+
- we trained our submission on 6 V100 32 GB GPUs provided by VSITE College for Information Technologies
- the inference took over a day on several GPUs (192 dense logits)



SWIFTNET: ROB 2020 (2)

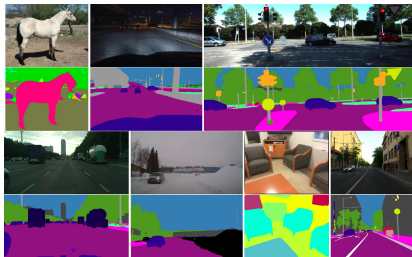
We achieved rank #1 on the semantic segmentation task:



Rank	Method	ACE2019 mIoU	Chaparral mIoU	RTTI mIoU	EPFL mIoU	Stanford mIoU	SWIFT mIoU	MIT mIoU
1	SWIFT	0.75	0.75	0.75	0.75	0.75	0.75	0.75
2	KingDL	0.74	0.74	0.74	0.74	0.74	0.74	0.74
3	Stanford	0.73	0.73	0.73	0.73	0.73	0.73	0.73
4	MIT	0.72	0.72	0.72	0.72	0.72	0.72	0.72
5	Stanford	0.71	0.71	0.71	0.71	0.71	0.71	0.71
6	Stanford	0.70	0.70	0.70	0.70	0.70	0.70	0.70
7	Stanford	0.69	0.69	0.69	0.69	0.69	0.69	0.69
8	Stanford	0.68	0.68	0.68	0.68	0.68	0.68	0.68
9	Stanford	0.67	0.67	0.67	0.67	0.67	0.67	0.67
10	Stanford	0.66	0.66	0.66	0.66	0.66	0.66	0.66

[orsic20arxiv]

The trained model can segment test images from multiple domains:



[bevandic22wacv]

SWIFTNET: WILDASH 2

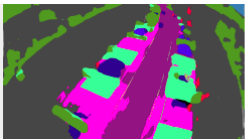
Our ROB 2020 model sets the state of the art on the most advanced road driving benchmark:

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

Algorithm	Meta AVG	Classic				Negative	Impact (IoU class)									
	IoU Class	IoU Class	IoU Class	IoU Cat.	IoU 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]

SWIFTNET: WILDDASH 2 (2)

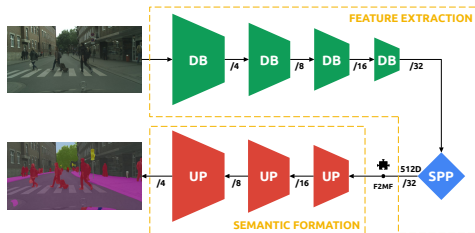
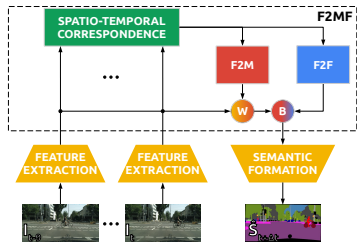


Part 2: Overview of current collaborations

COLLABORATION: RIMAC AUTOMOBILI

Dense semantic forecasting [saric20cvpr]:

- guess the content of short-term (120 ms) and mid-term (540 ms) future images
- give up upsampling to forecast single-level low-resolution features
- the approach can be applied to three dense prediction tasks



[saric20cvpr]

COLLABORATION: RIMAC AUTOMOBILI (2)

Some experimental results:

- model correctly forecasts disappearance of the car
- blue pixels indicate that our model is aware that this part of the road has to be imagined



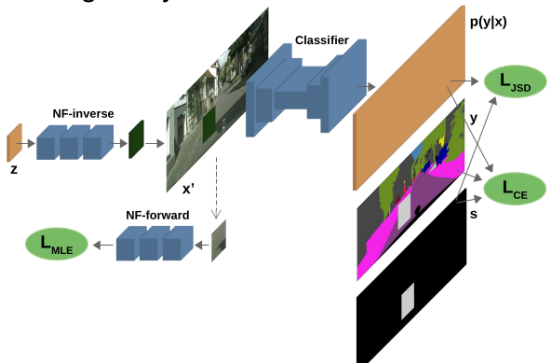
[saric20cvpr]

Accuracy (mIoU)	Short term: $\Delta t=3$		Mid term: $\Delta t=9$	
	All	MO	All	MO
Oracle-DN121	75.8	75.2	75.8	75.2
Oracle-RN18	72.5	71.5	72.5	71.5
Copy last (DN121)	53.3	48.7	39.1	29.7
3Dconv-F2F [39]	57.0	/	40.8	/
Dil10-S2S [11]	59.4	55.3	47.8	40.8
LSTM S2S [33]	60.1	/	/	/
Mask-F2F [12]	/	61.2	/	41.2
FeatReproj3D [40]	61.5	/	45.4	/
Bayesian S2S [32]	65.1	/	51.2	/
LSTM AM S2S [34]	65.8	/	51.3	/
LSTM M2M [14]	67.1	65.1	51.5	46.3
F2MF-RN18 w/o d.a.	66.9	65.6	55.9	52.4
F2MF-DN121 w/o d.a.	68.7	66.8	56.8	53.1
F2MF-DN121 w/ d.a.	69.6	67.7	57.9	54.6
F2MF-DN121 w/ d.a. \dagger	70.2	68.7	59.1	56.3

COLLABORATION: GIDEON BROTHERS

Open-set recognition by training on synthetic negative images:

- mixed-content training with pasted negative patches
- the negatives are sampled from a jointly trained generative model
- state-of-the-art performance on Segment Me If You Can
- better than all other approaches which do not use real negatives and image resynthesis.



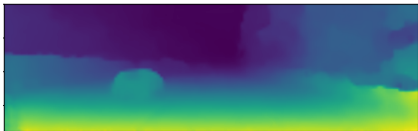
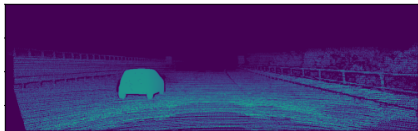
COLLABORATION: GIDEON BROTHERS (2)

Self-supervised learning of stereoscopic reconstruction [liu20cvpr]:

- the training proceeds on neighbouring stereo pairs
- self-supervised geometrical constrains, eg:
 $\text{flow}(l1, l2) + \text{stereo}(l2, r2) = \text{flow}(l1, r2)$



[geiger13ijrr]

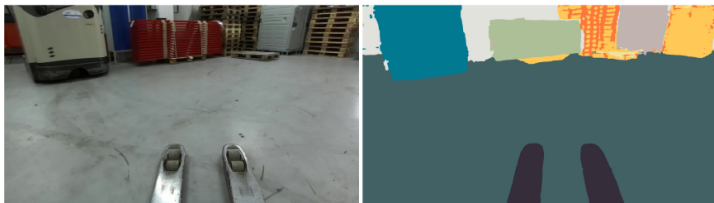


[sovic21]

COLLABORATION: ROMB

Full-stack solution for embedded real-time perception:

- management of the labeling process [orsic21phd]
- dense prediction with pyramidal SwiftNet [orsic21pr]
- training suitable non-ImageNet initialization [bevandic22wacv]
- semi-supervised training [grubisic21mva]
- optimization with TensorRT on Jetson AGX [orsic21phd]
- optimization through static quantization [orec21ms]
- pruning according to the lottery ticket hypothesis [bratulic22]



COLLABORATION: MICROBLINK

We address sparse monocular reconstruction.

- focus on self-supervised correspondence
- require near-real-time performance on mobile devices.



[orsic21]

COLLABORATION: UNIZG-FPZ

Multi-label classification in video:

- around 40 attributes related to road safety
- knowledge transfer from our state-of-the-art segmentation models
- clear commercial potential, struggle with dataset imbalance and noisy labels
- current work considers monocular reconstruction and panoptics



[kaca20itsc]

CHALLENGES AND OPPORTUNITIES

Part 3: Challenges and Opportunities

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
- we know we could improve faster with more computational power
- hardware improves slowly and wastes a lot of energy.

Clearly, there is some uncertainty ahead.

However, catastrophic forecasts appear largely exaggerated.

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: 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 : PROSPECTS

The prospects for young researchers are quite good.

- competent employers who require relevant competence
- decent pays for interesting jobs
- world is still quite open
- most research papers have corresponding github pages

CONCLUSIONS : SPACE FOR IMPROVEMENT

Two things could be improved.

1. compulsory journal publications

- our decision delays distribute according to $\mathcal{N}(13, 3)$ (months)
- if one is unlucky, (s)he can be rejected after 15 months of review, and then having to wait for another 15 months
- outcome: promotion of poor journals with large APCs instead of top conferences

2. availability of GPU power

- many research papers inaccessible in spite of published code.
- Isabella is insufficient and crowded with jobs with very low GPU utilization.
- we should think about funding a computing center with at least 12-fold performance of Isabella.

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 (MULTICLOUD, ADEPT), ERDF (DATACROSS, A-UNIT, SAFETRAM), Rimac automobili, Microblink, Gideon brothers, Romb technologies, Končar, UniZg-FPZ, and VSITE.