Supervised convolutional models for natural image understanding in road traffic

Siniša Šegvić, Ivan Krešo, Josip Krapac
Faculty of Electrical Engineering and Computing
University of Zagreb

Introduction: about

Semantic segmentation: image understanding at the pixel-level

- pixel-level: associate each pixel with a class
- image understanding: classes have a high-level meaning

  traffic participants: person (red), car (blue), bicycle (dark red)

  objects: pole (light grey), traffic sign (yellow), traffic light (orange)

  landscape: road (purple), sidewalk (pink), building (dark grey), vegetation (dark green), terrain (light green), sky (light blue)






Introduction: agenda

1. About semantic segmentation
   - overview, fully convolutional approach, problems
2. Achieving scale invariance by depth-driven selection
   - improve recognition with reconstruction
3. Restoring the resolution with ladder-style upsampling
   - blend semantic information with spatial accuracy
   - recover semantic information with the DenseNet architecture
4. Experiments
   - performance criteria, datasets, results
5. Conclusion

Overview: recognition

Example: discriminate bison from oxen

[image-net.org]

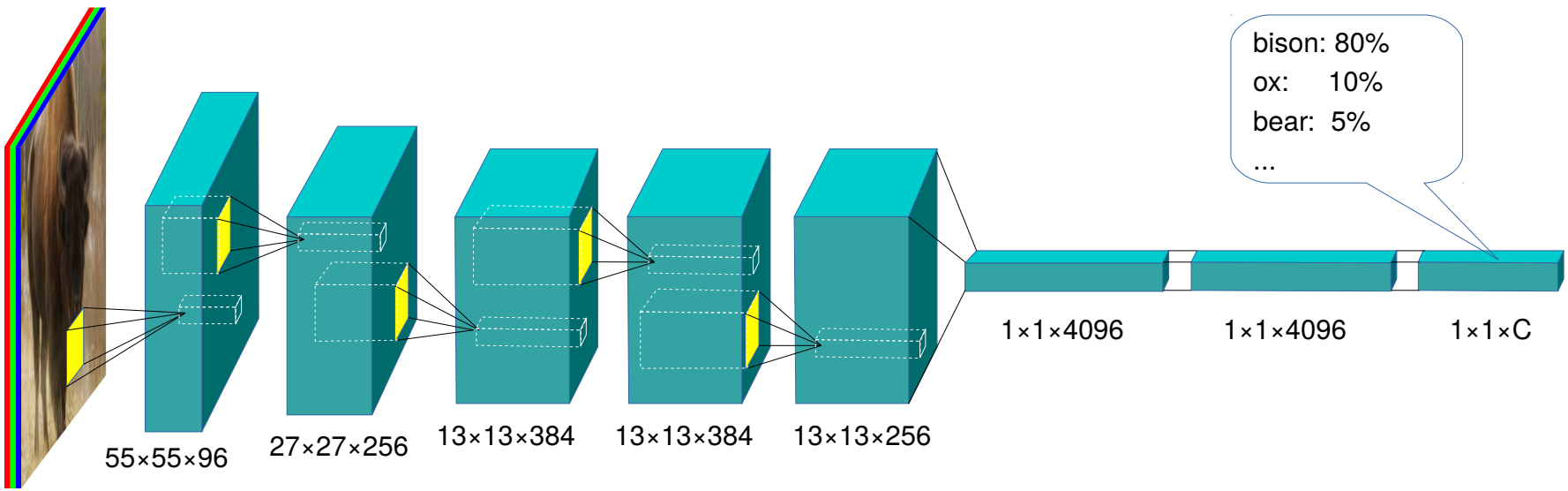
1. express the program with many free parameters
   - the parameters determine a transformation which we call the model
2. fit parameters on the training set
3. evaluate performance on the test set

Success depends on the model, training set and processing power.

Overview: architecture

Deep convolutional model for image classification [krizhevsky12nips]

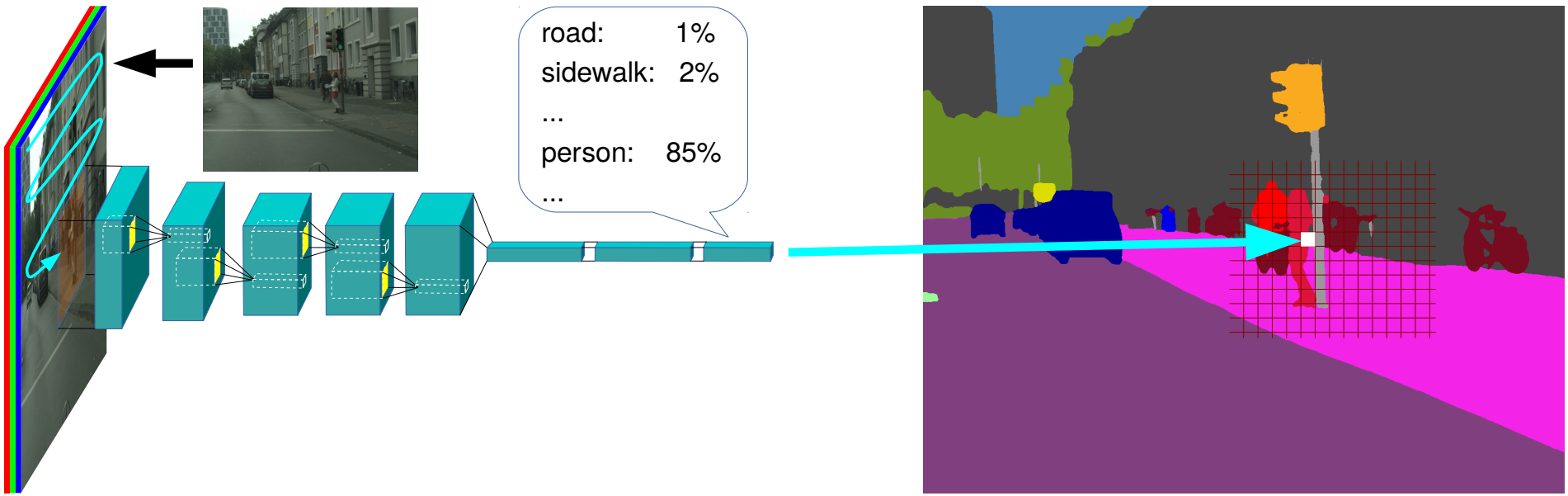
- **input**: image; **output**: distribution over 1000 classes
- **fitness criterion**: average log probability of the correct class
- **structure**: a succession of convolutions and poolings
  - gradual decrease of resolution and increase of the semantic depth
- recent architectures: O(10^2) layers, O(10^6) parameters, O(10^9) multiplications for a 224x224 image!



Overview: sliding window

A classification model can be applied to the segmentation task:

- analyze the image in the **sliding window** fashion
  - each patch produces one pixel of the semantic map
- segmentation groundtruth allows end-to end training
- each pixel becomes one component of the fitness criterion
- optimized implementation required in practice
  - 10^6 pixels $\times$ 10^9 multiplications?

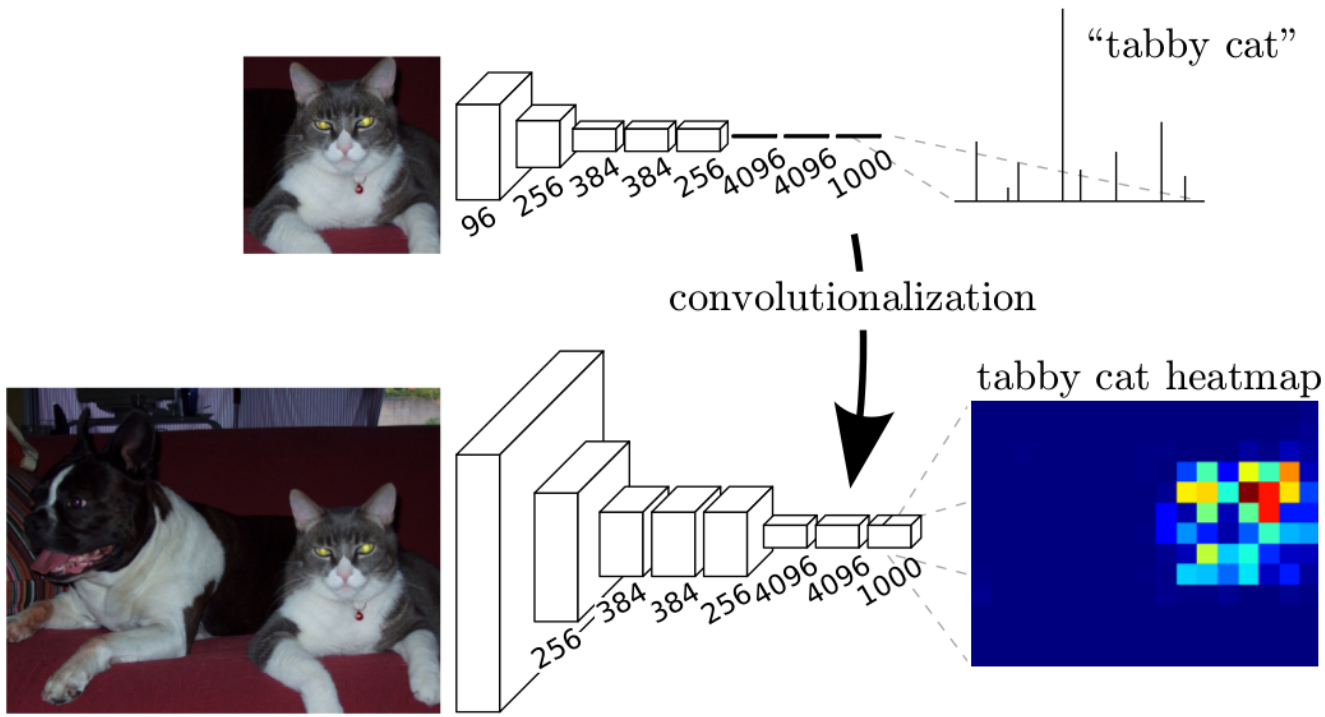


Overview: going fully convolutional

Luckily, the processing of neighbouring patches involves calculating many common latent activations

More efficient: perform the classification layer-wise [long15cvpr]:

- the resulting semantic map is subsampled due to pooling
- this can be relaxed to some extent with dilated filtering [yi16iclr]

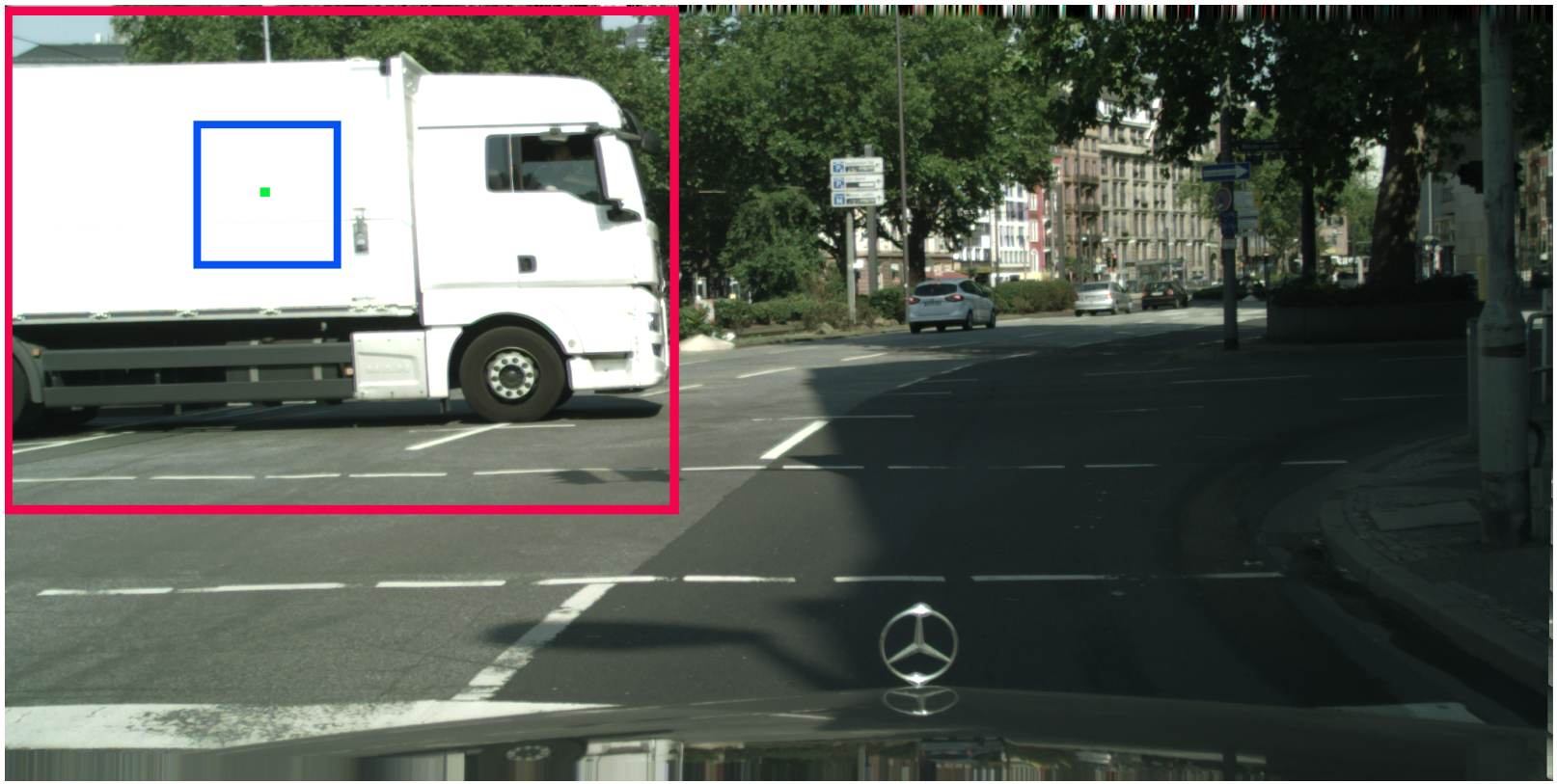


[long15cvpr]

Problems: large objects

Classifying pixels at large objects may require a huge receptive field.

- many local neighbourhoods are not discriminative enough
- their center pixel can only be recognized in a larger context
- problem arises when the context is larger than receptive field

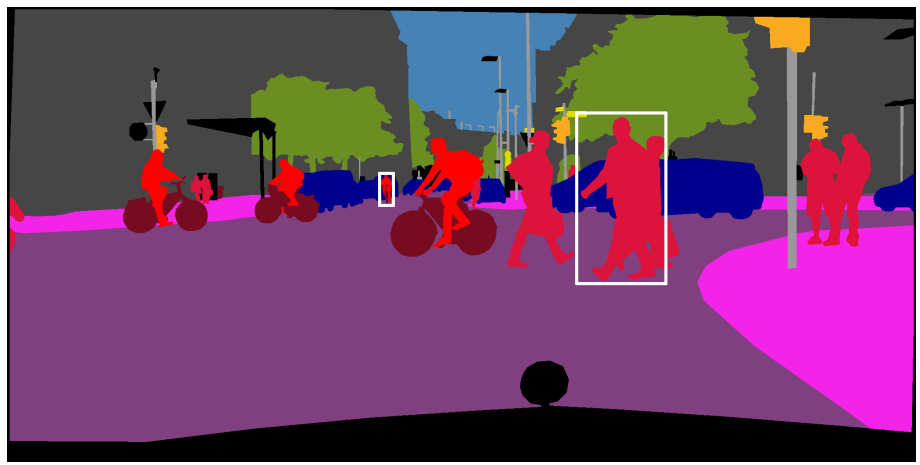


[kreso16gcpr]

Problems: small objects

Detecting small objects with a huge receptive field wastes resources:

- small and simple objects can be recognized with few layers
- latter layers forward their activations over and over again
- that leads to loss of the representational power

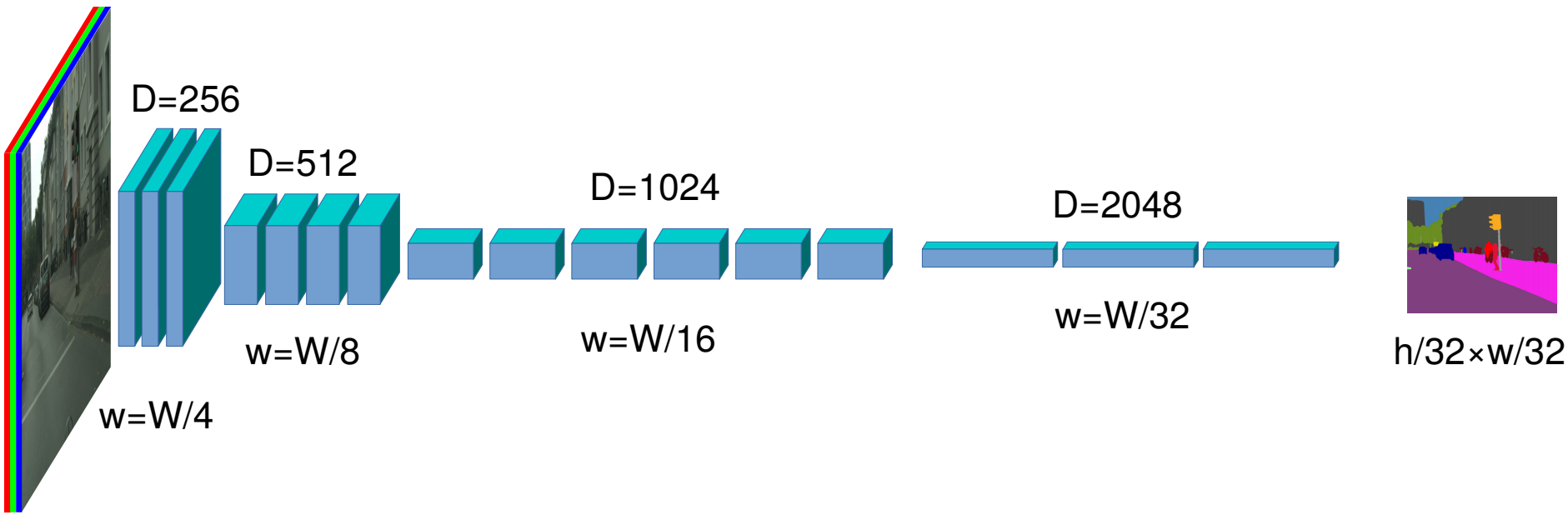


[kreso16gcpr]

Problems: memory requirements

Successful segmentation architectures rely on pretrained models designed for ImageNet: small input resolution, single pixel output

- in segmentation we have large resolution both on input and output
- brute force output restoration (dilated filtering) is feasible up to 8x ↑



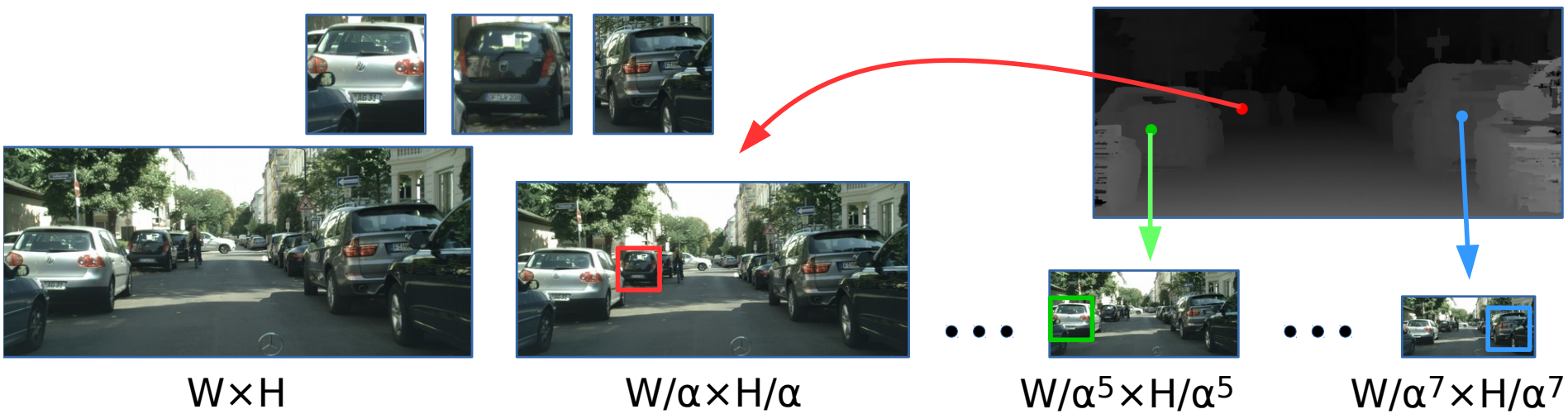
Mainly relevant for training, where all activations must be cached

- however, lean memory requirements favour short evaluation times

Scale invariance: idea

Use stereo reconstruction to disentangle appearance from scale

- independently extract features from all levels of **image pyramid**
- analyze each pixel at the pyramid level determined by its **distance** from the camera
- **effect**: image objects are perceived at the common scale regardless of the camera distance

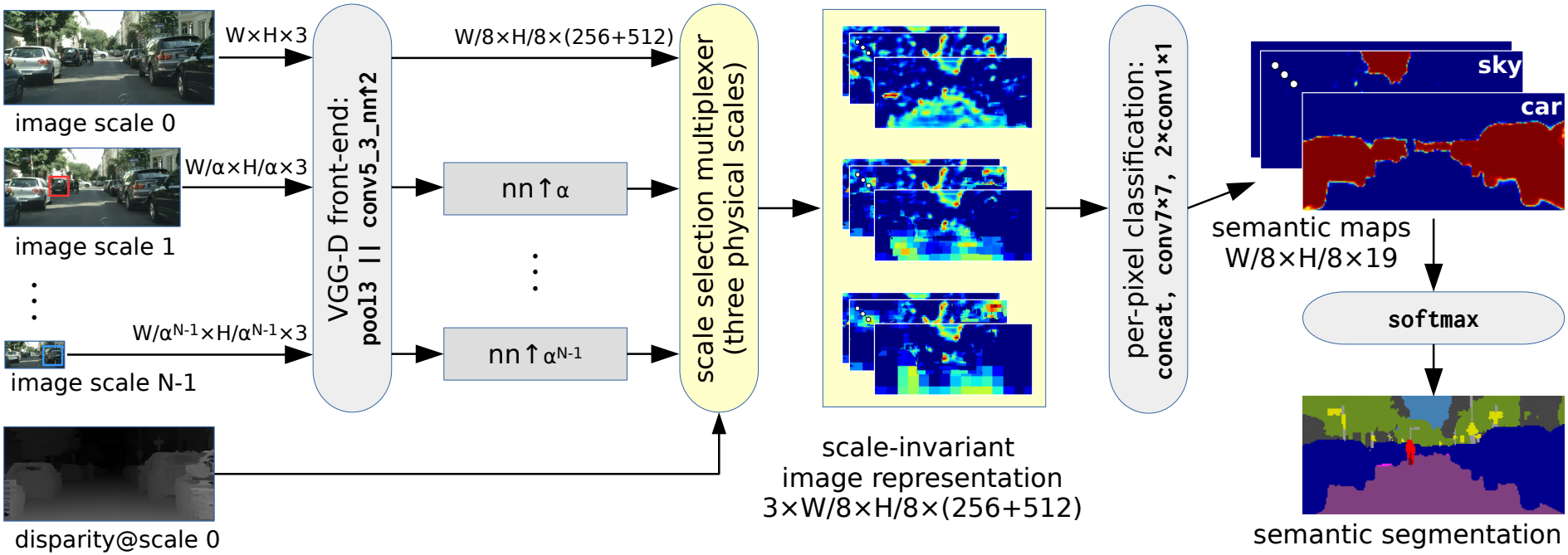


[kreso16gcpr]

Scale invariance: architecture

We introduce a new layer: the scale selection multiplexer

- the multiplexer assembles pieces from appropriate pyramid levels
- the back-end receives a **scale-invariant** feature mosaic
- the scale multiplexer is compatible with **end-to-end** training

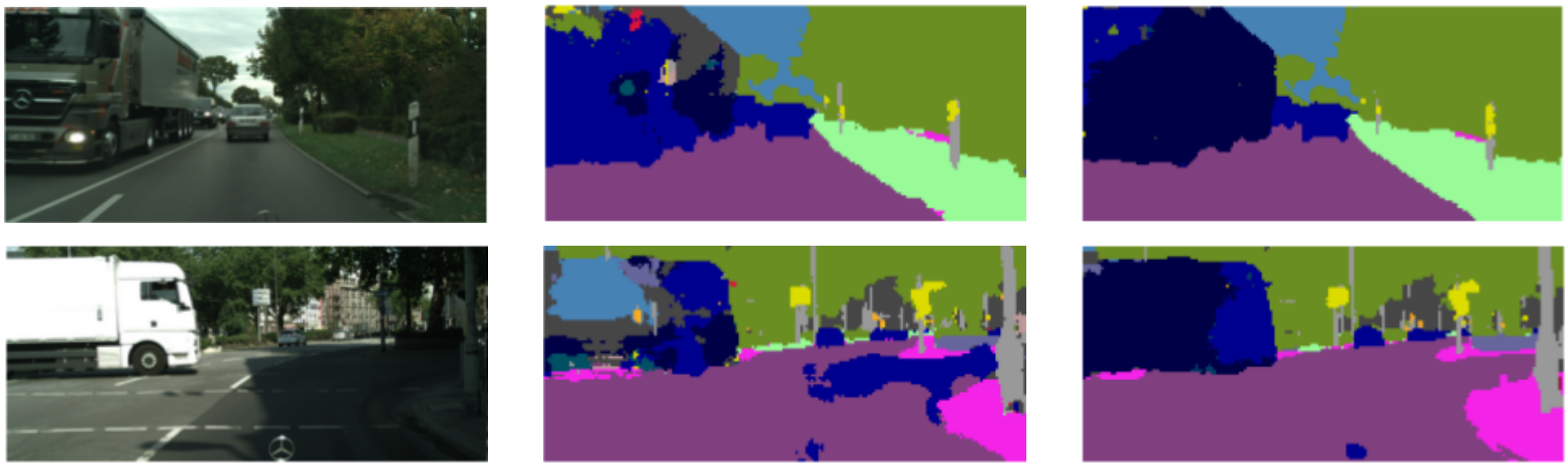


[kreso16gcpr]

Scale invariance: results

Scale selection solves the problems of large and small objects:

- nearby objects are recognized in diminished images
- far objects are recognized at the original resolution



Effects of scale selection (mIoU): 56.4 $\rightarrow$ 64.4

Predictions are still subsampled, the memory problem remains

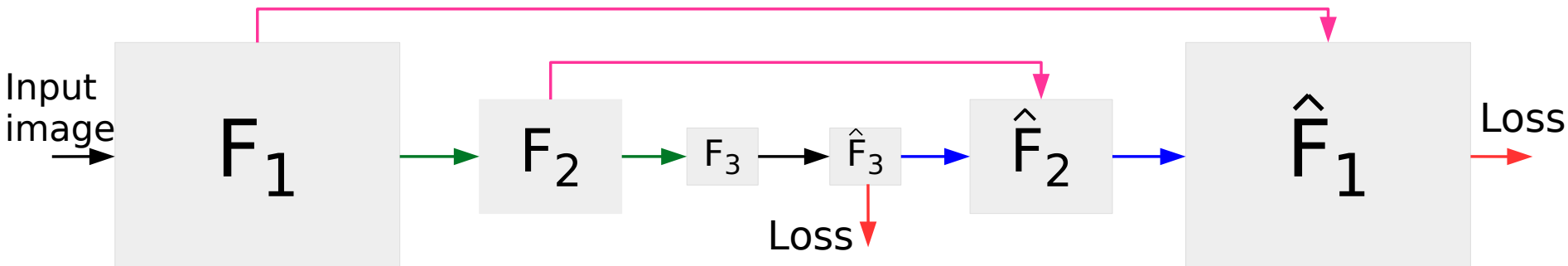
Smart upsampling: idea

Using a full-fledged ImageNet-class model in each pixel is wasteful:

- boundary refinement should be easier than recognition
- upsample a deep representation by blending it with a higher resolution earlier layer [valpola14arxiv,ronneberger15arxiv]

The resulting architecture operates as follows:

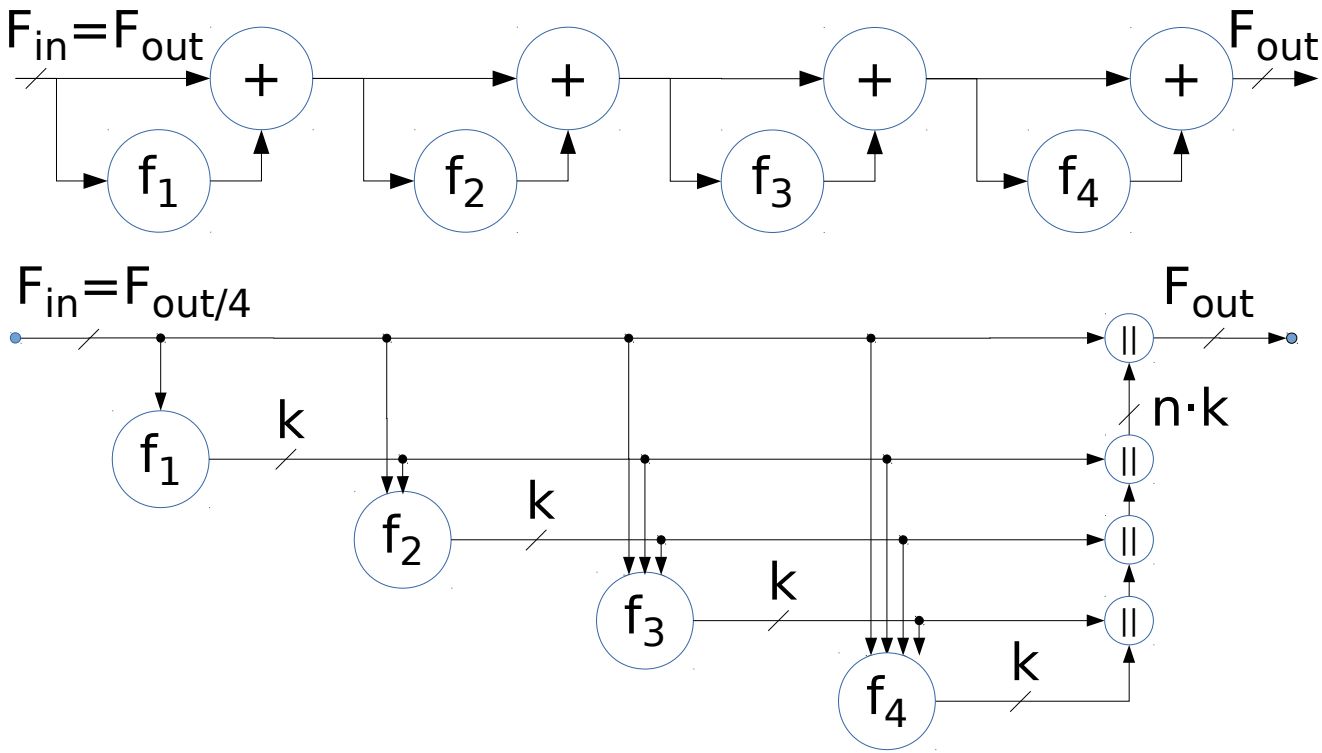
- the downsampling datapath infers the semantic information
- the upsampling datapath refines the boundaries
- the lateral connections ensure the blending: $\hat{F}_t = g_t(F_t, \hat{F}_{t+1})$



Smart upsampling: DenseNet

The downsampling datapath can be any classification architecture

- we compare ResNet [he16cvpr] and DenseNet [huang17cvpr]

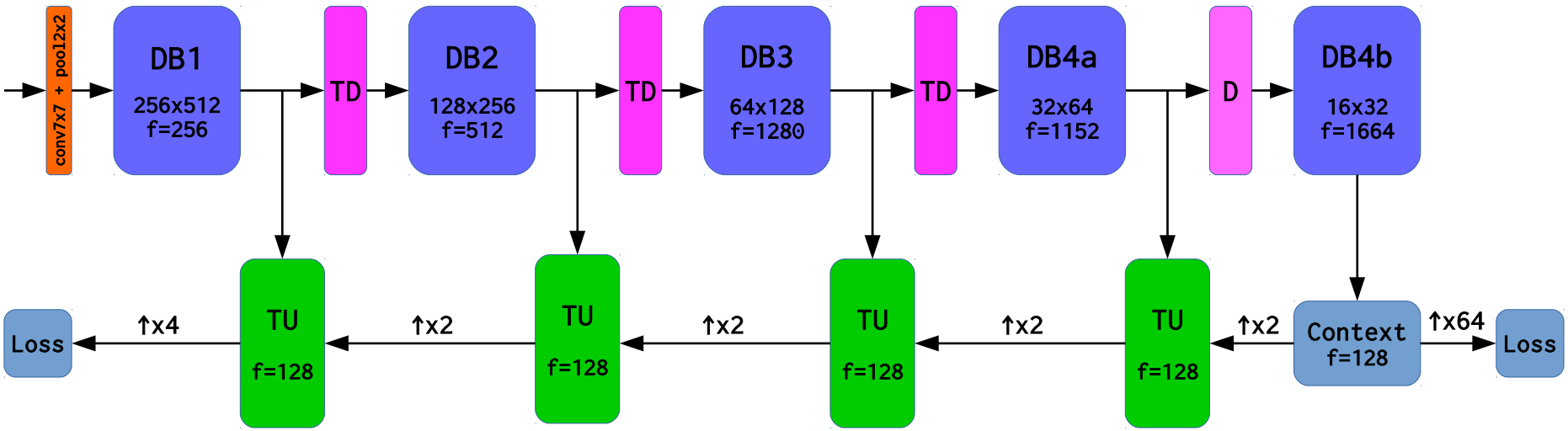


- both architectures favour the gradient flow towards the **early layers**
- **hypothesis**: DenseNet is better when the class complexities vary

Smart upsampling: ladder-style

Hypothesis: recognizing objects is harder than boundary refinement

- use a **lean** representation in the upsampling datapath
- this results in huge **memory savings**, unlike [jegou16arxiv]



We split DB4 into DB4a and DB4b to increase receptive field

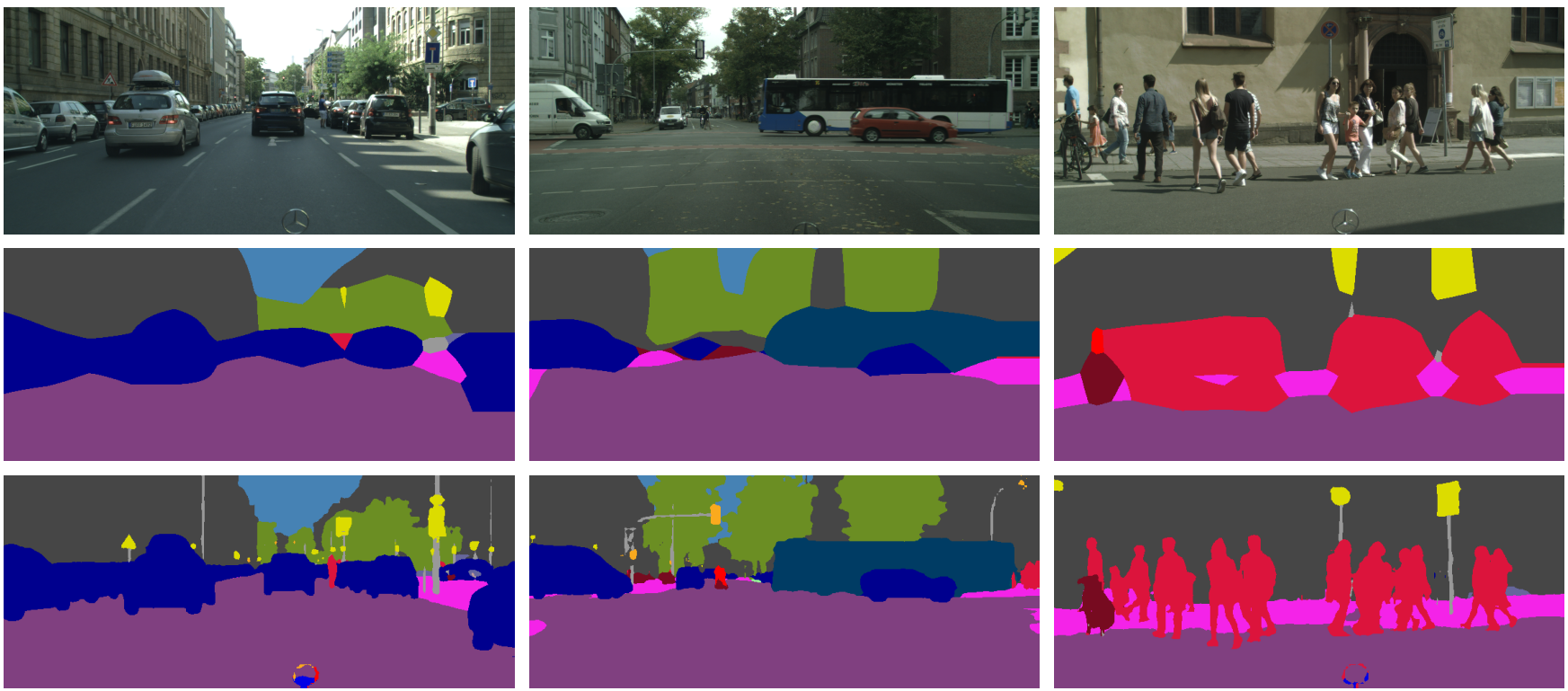
- this results in 64$\times$ downsampling (appropriate for large objects)
- this compromises ImageNet initialization for DB4b
- however, **fine-tuning** succeeds to recover

Smart upsampling: results

We train on full Cityscapes resolution with bs=2 on two GTX1070

We recover fine details lost due to 64× downsampling

- middle: upsampling 64× by interpolation
- bottom: upsampling 16× by blending and 4× by interpolation

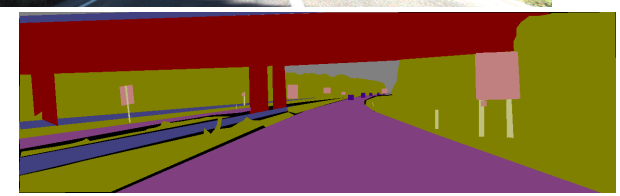


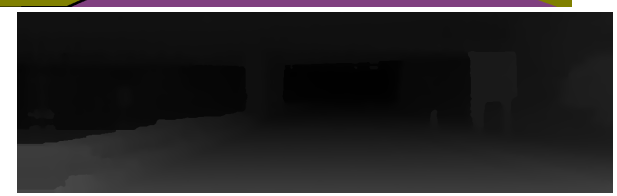
Effect of blending (mIoU): 62.5 → 72.8

Evaluation: datasets

- Pascal VOC 2012x [everingham10ijcv]:
  - generic photographs
  - 10 indoor, 10 outdoor classes
  - 12000 images, <.25 MPixel

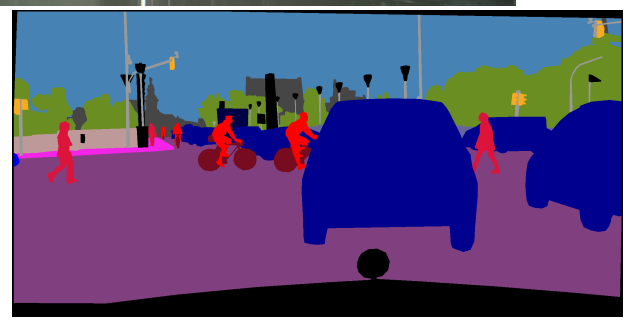
- KITTI [geiger13ijrr]:
  - driver's perspective, 11 classes
  - 450 stereo images, 0.5 MPixel
  - reconstruction groundtruth
  - odometry groundtruth
  - Karlsruhe, fine weather



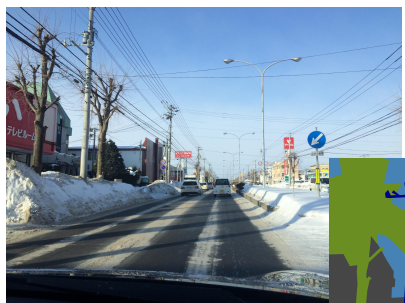


Evaluation: datasets (2)

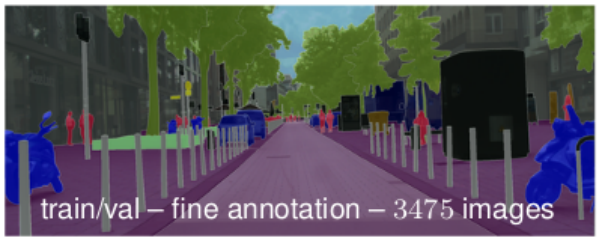
- Cityscapes [cordts16cvpr]:
  - driver's perspective, 19 classes
  - 5000 stereo images, 2MPixel
  - 20000 coarsely annotated images
  - instance level annotations
  - 50 cities, spring to autumn

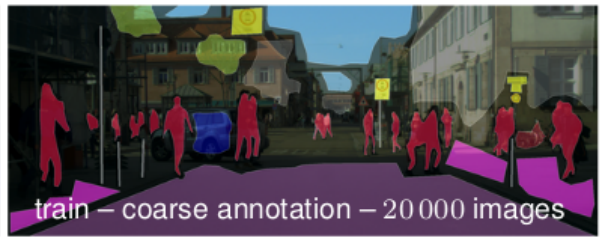


- Vistas [neuhold17iccv]:
  - driver's perspective, 100 classes
  - 25000 images, 2-8 MPixel
  - instance level annotations
  - worldwide, various weather

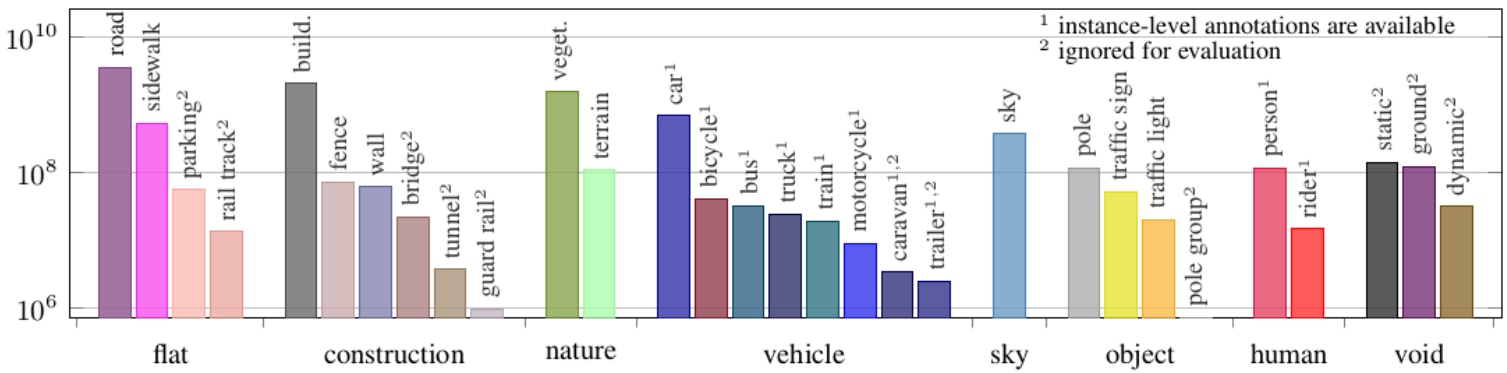


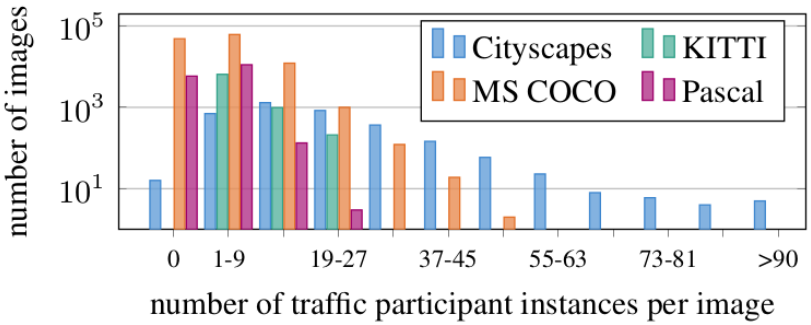
Evaluation: cityscapes [cordts16cvpr]

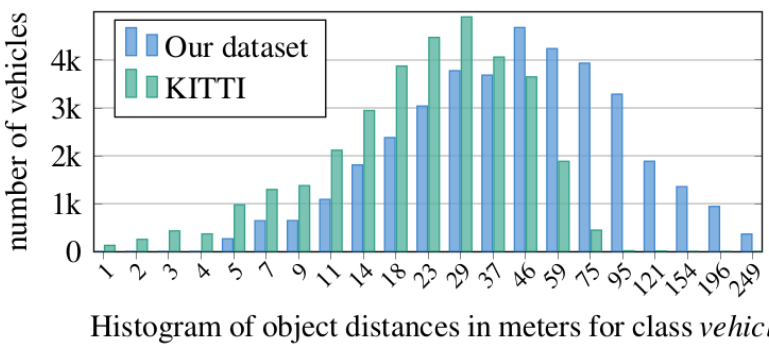




Pros: fine annotation, many classes, well-chosen categories, complex cluttered scenes, variety of scale



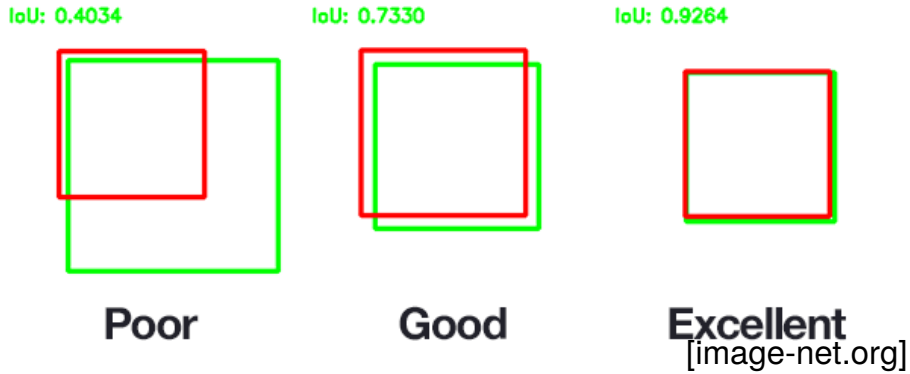




Evaluation: performance characterization

Widely used performance metric: intersection over union (IoU)

- set A: groundtruth pixels of class c
- set B: predicted pixels of class c
- $\text{IoU}_X = |A \cap B| \,/\, |A \cup B|$



[image-net.org]

Typically, the performance is expressed as mean IoU over all classes

- $\text{mIoU} = \frac{\sum_c \text{IoU}_c}{C}$
- this increases the influence of rare classes with few training pixels
- examples: wall, fence, pole, bottle, potted plant

To ensure integrity, labels of test subsets are withheld from public datasets

The performance on the test set is determined by submitting results to the evaluation server

Evaluation: test errors

bus/truck
road/sidewalk

tram/bus
sidewalk/road
pole/sign

bus/car/building
sidewalk road
pedestrian/rider

Evaluation: test errors (2)

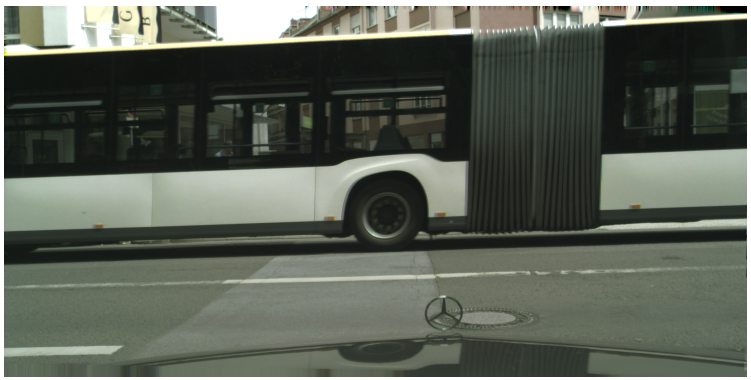
truck/car
motorcycle/bicycle
building/wall
building/sign

bus/building/truck
pedestrian/rider
car/building

tram/bus/truck
road/sidewalk

Evaluation: test errors (3)

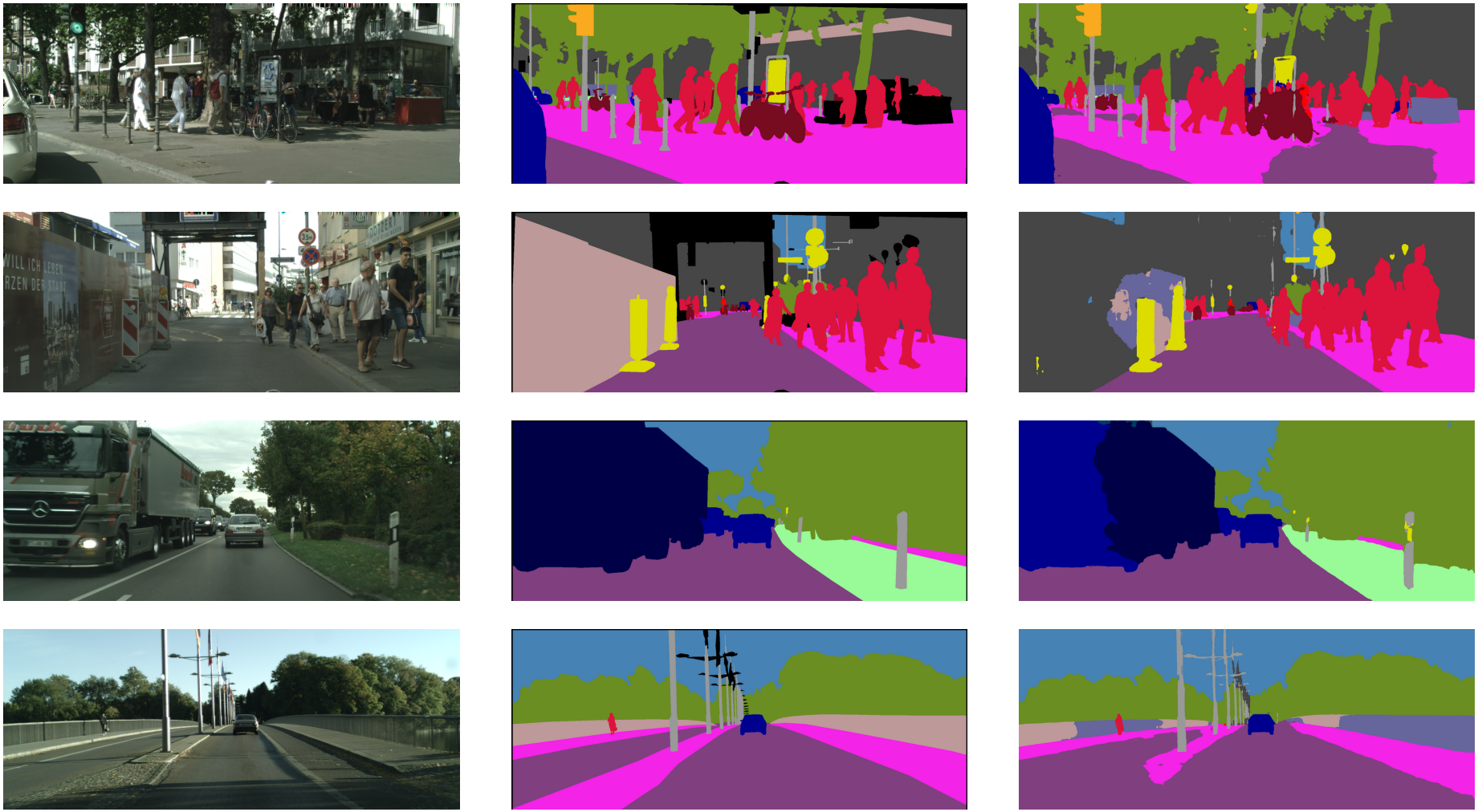
bus/tram
sidewalk/road



bus/tram/car
bus/truck/wall/fence

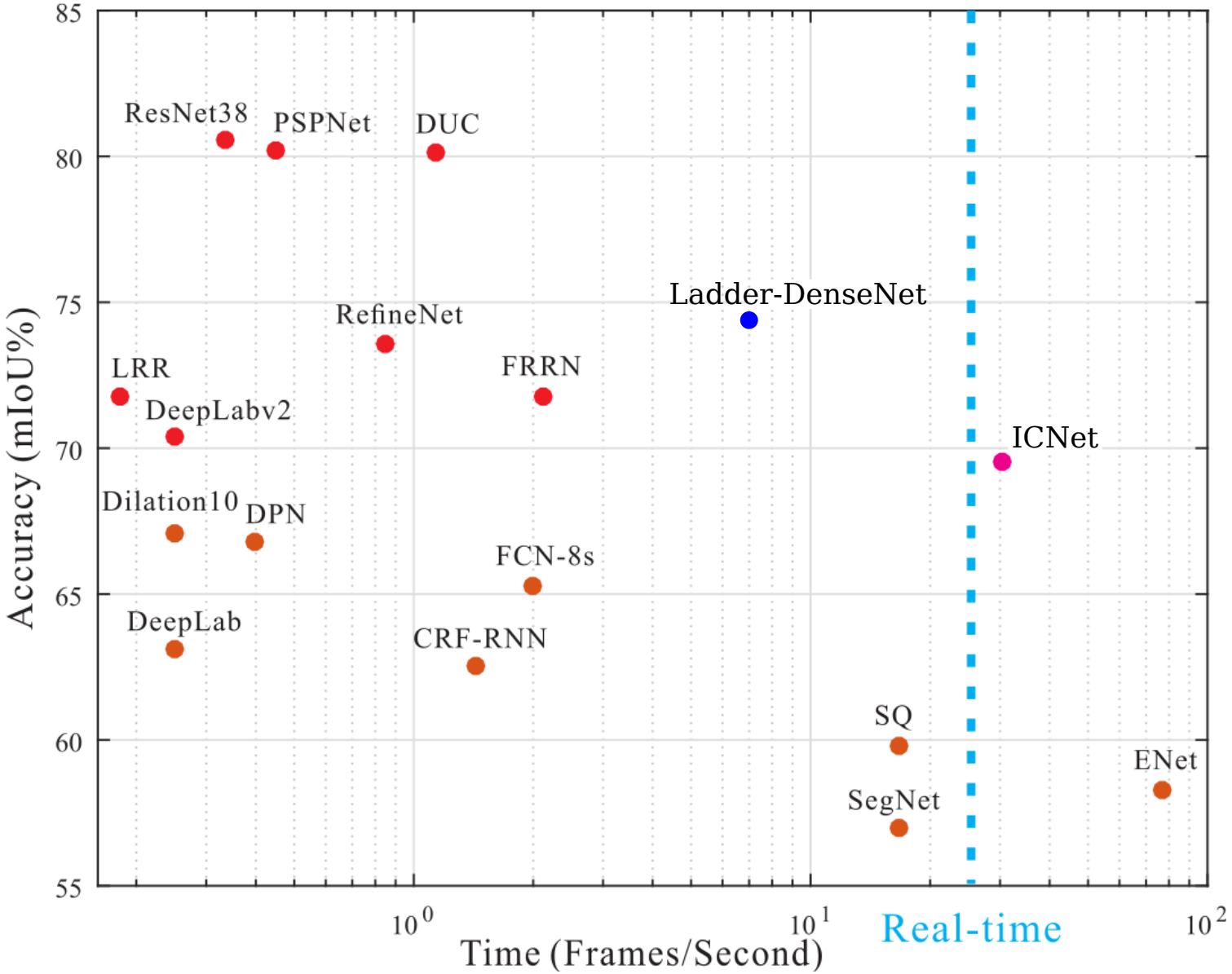
car/truck/tram/
car/building/sign

Evaluation: worst performance on val



Problems: i) ambiguous sidewalk, ii) unusual fence, iii) large truck, iv) fence vs wall, occlusions.

Evaluation: performance



[zhao17arxiv]

Conclusion

Scale selection: Cityscapes mIoU 56.4 $\rightarrow$ 64.4

Ladder-style upsampling (mIoU): Cityscapes 62.5 $\rightarrow$ 72.8

ResNet vs DenseNet: Cityscapes 69.5 $\rightarrow$ 72.8; VOC12 63.0 $\rightarrow$ 70.2

Performance on test (mIoU): Cityscapes 74.6; VOC 2012 AUG 78.0

Able to train at full Cityscapes resolution with bs=2 on 2$\times$GTX1070

Able to process 1024$\times$448 images at 31Hz with a 74.6 mIoU model

Able to recover from 64$\times$ subsampling

Conclusion: discussion

Thank you for your attention!

Questions?

This presentation has been fully supported by Croatian Science Foundation under the project I-2433-2014.
http://multiclod.zemris.fer.hr

The Titan X used in experiments was donated by NVIDIA Corporation.

Appendix: adversarial examples