Deep learning for semantic segmentation

Andra Petrovai

Research Center for Image Processing and Pattern Recognition

Technical University of Cluj-Napoca

2017 IEEE International Conference on Intelligent Computer Communication and Processing
September 7-9, Cluj-Napoca, Romania

Contents

- Introduction
- Benchmarks
- FCN
- ERF-Net
- Transfer learning

Semantic segmentation

- Label each pixel in the image with a semantic class
- Don't differentiate between instances
- Provides a detailed understanding of the environment
- Can be used in the context of autonomous driving

Benchmarks

- Various **indoor and outdoor scenes** and images of **objects, persons and animals**:
- **Pascal VOC** – 21 classes, 10k images [1]
- **Pascal Context** – 59 classes, 10k images [2]
- **Microsoft COCO** – 182 classes, 10k images [3]
- **ADE20K** – 150 classes, 20k images [4]

Benchmarks

- Contain only **traffic scenes**:
- **CamVid** – 32 classes, 700 images [5]
- **Cityscapes** – 30 classes, 19 classes used in evaluation, 5000 images [6]
- **Synthia –** 13 classes, 13.400 images, synthetic dataset [7]

Fully Convolutional Network

- Fully Convolutional Network [8]: a special type of convolutional network that outputs a segmented image
- Convolution network: downsampling
- Deconvolution network: upsampling

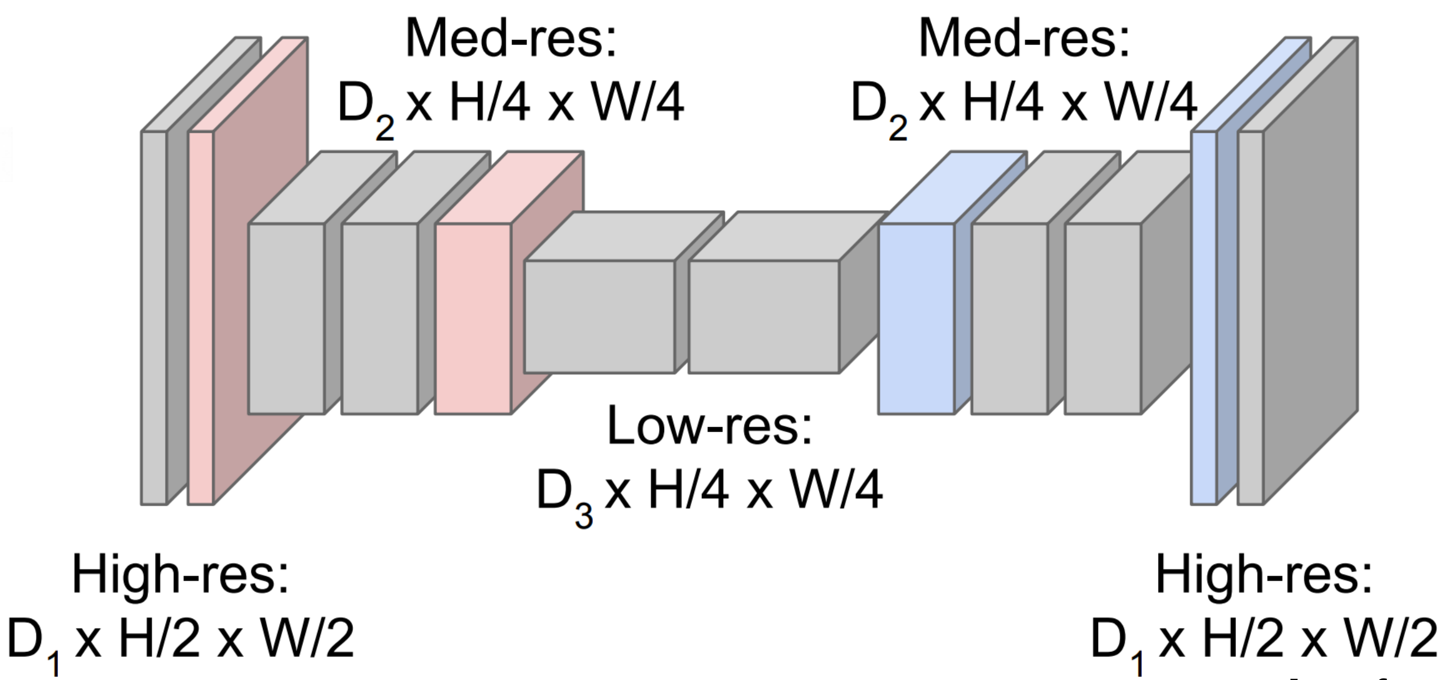


Figure taken from FCN paper [8]

FCN: Max Unpooling

1	2	6	3
3	5	2	1
1	2	2	1
7	3	4	8

Input: 4 x 4

Max pooling with 2x2 filters and stride 2

5	6
7	8

Output: 2 x 2

Rest of the network

1	2
3	4

Input: 2 x 2

Max unpooling (use positions from the pooling layer)

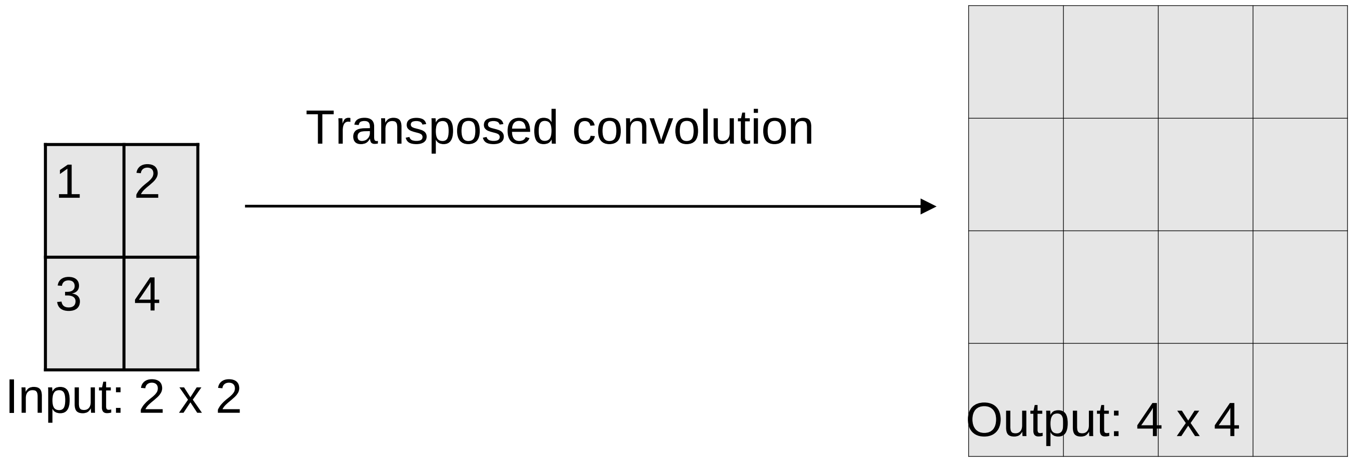
0	0	2	0
0	1	0	0
0	0	0	0
3	0	0	4

Output: 4 x 4

- Use corresponding pairs of pool/unpooling layers

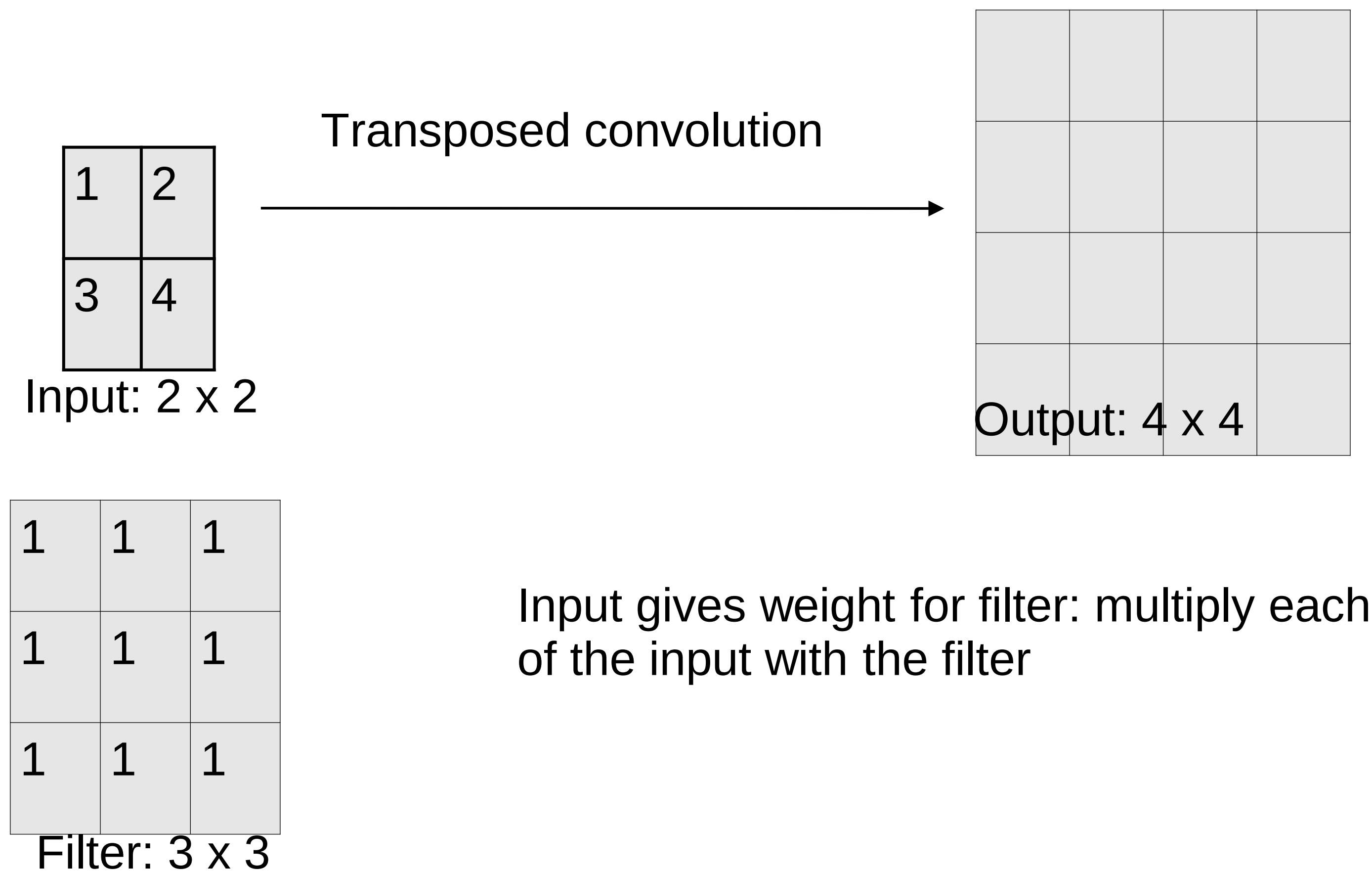
FCN: Transposed Convolution

- 3 x 3 transposed convolution, stride 2 pad 1



FCN: Transposed Convolution

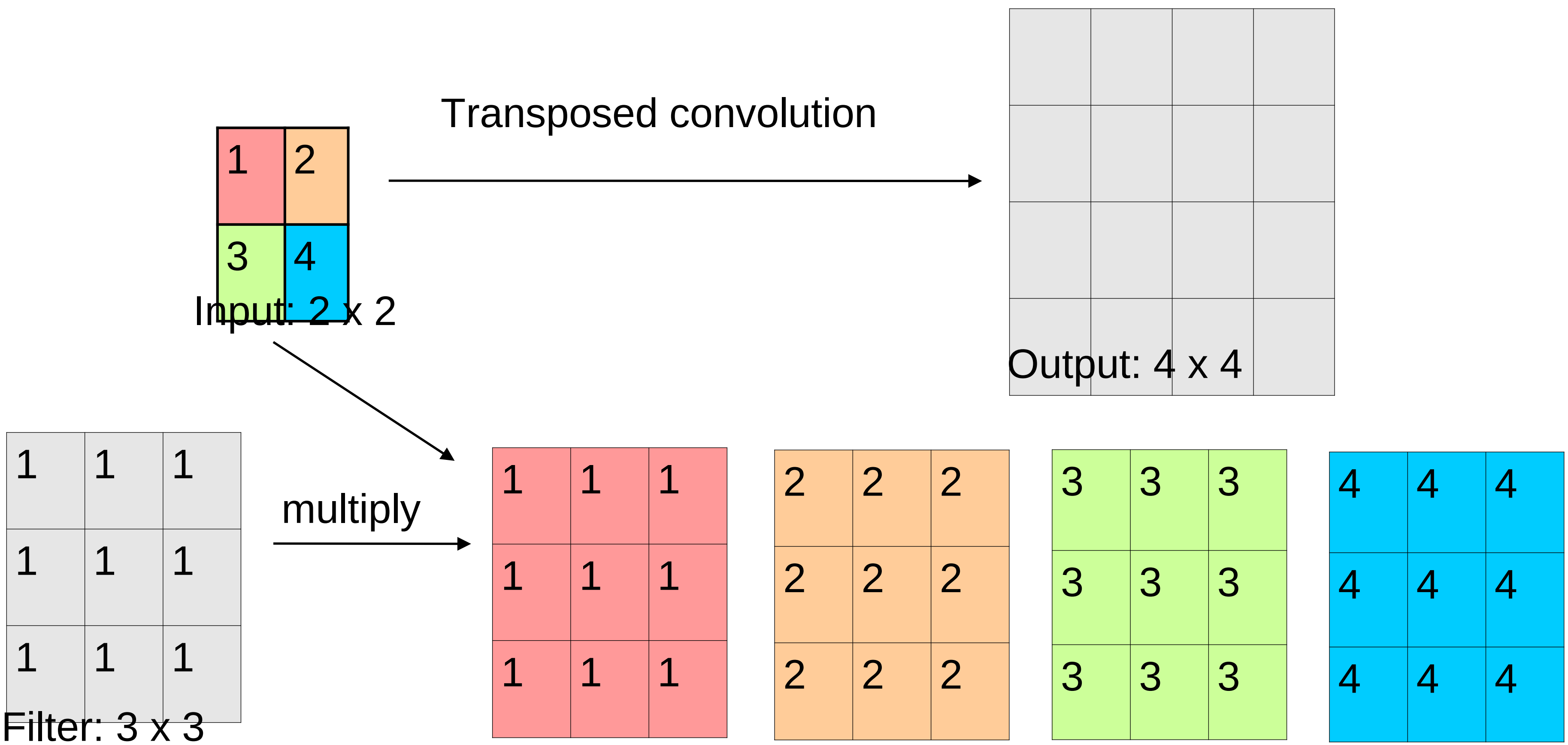
- 3 x 3 transposed convolution, stride 2 pad 1



Input gives weight for filter: multiply each element of the input with the filter

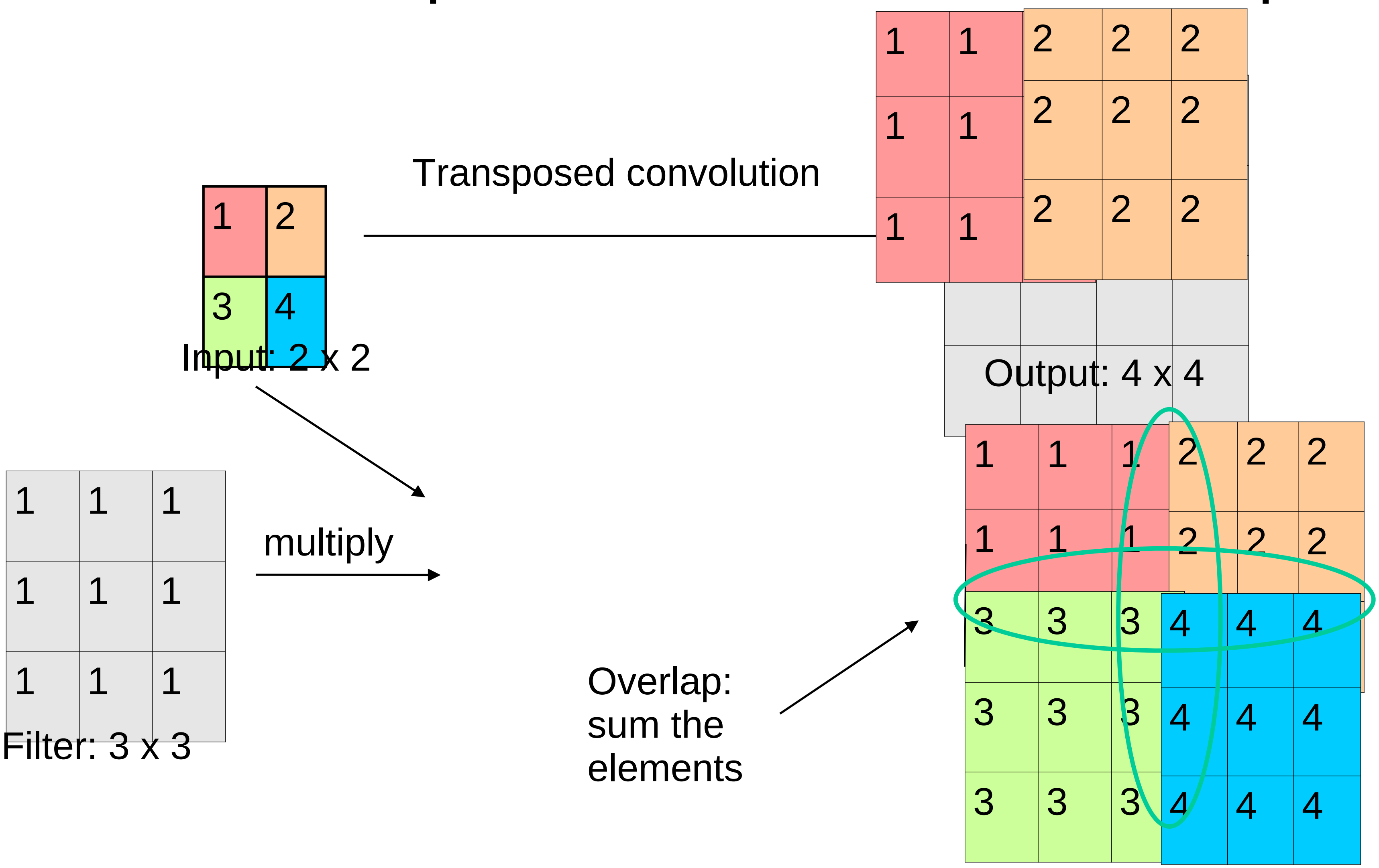
FCN: Transposed Convolution

- 3 x 3 transposed convolution, stride 2 pad 1



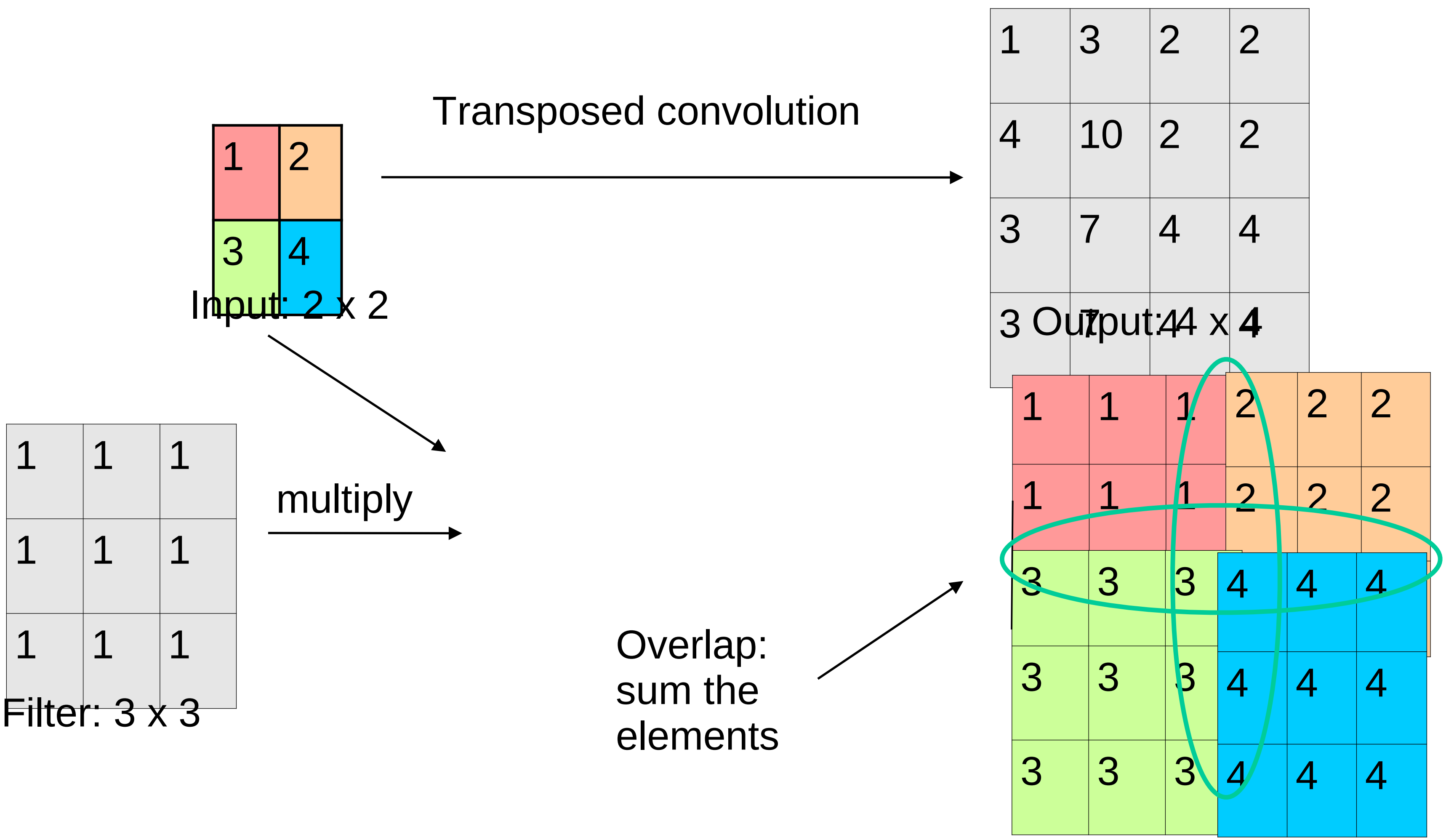
FCN: Transposed Convolution

- 3 x 3 transposed convolution, stride 2 pad 1



FCN: Transposed Convolution

- 3 x 3 transposed convolution, stride 2 pad 1



FCN performance on the Cityscapes dataset

name	fine	coarse	16-bit	depth	video	sub	IoU class	iIoU class	IoU category	iIoU category	Runtime [s]	code
m-TCFs	yes	yes	no	no	no	no	71.8	43.6	87.6	70.6	1.0	no
LRR-4x	yes	yes	no	no	no	no	71.8	47.9	88.4	73.9	n/a	yes
FRRN	yes	no	no	no	no	2	71.8	45.5	88.9	75.1	n/a	yes
Adelaide_context	yes	no	no	no	no	no	71.6	51.7	87.3	74.1	n/a	no
ML-CRNN	yes	no	no	no	no	no	71.2	47.1	87.7	72.5	n/a	no

...

Position 22/55

name	fine	coarse	16-bit	depth	video	sub	IoU class	iIoU class	IoU category	iIoU category	Runtime [s]	code
FCN 8s	yes	no	no	no	no	no	65.3	41.7	85.7	70.1	0.5	yes

ERF-Net [9]

- One of the fastest convolutional neural networks in the literature used for semantic segmentation
- Execution time is **25 ms** for a **1024 x 512** image running on a Titan X GPU

ERF-Net: Architecture

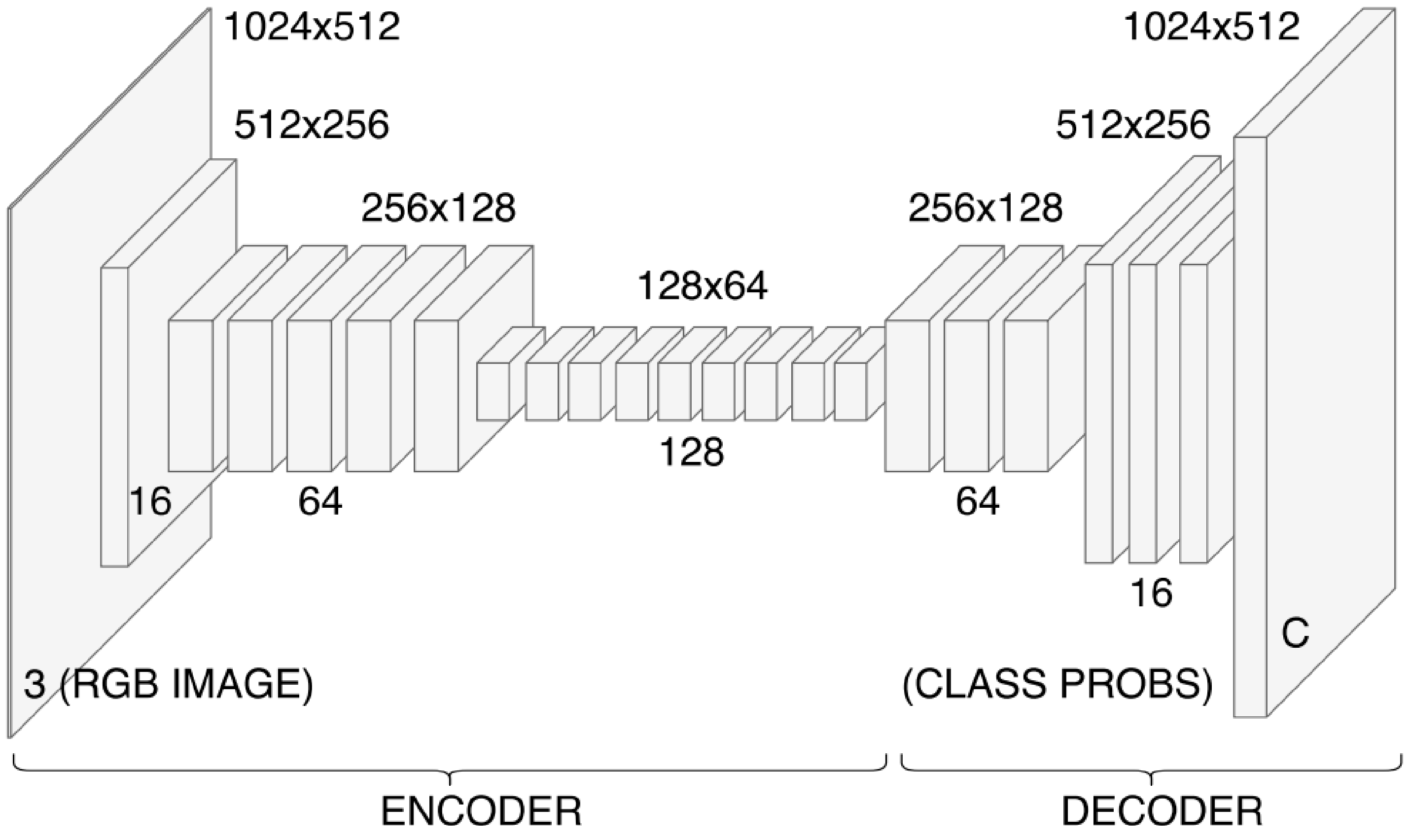


Figure taken from paper [9]

ERF-Net: Architecture

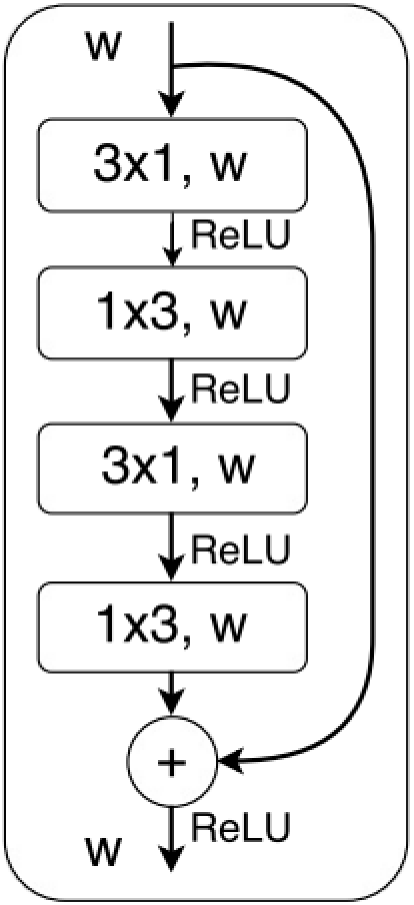
Layer	Type	out-F	out-Res
1	**Downsampler block**	16	512x256
2	**Downsampler block**	64	256x128
3-7	5 x **Non-bt-1D**	64	256x128
8	**Downsampler block**	128	128x64
9	**Non-bt-1D** (dilated 2)	128	128x64
10	**Non-bt-1D** (dilated 4)	128	128x64
11	**Non-bt-1D** (dilated 8)	128	128x64
12	**Non-bt-1D** (dilated 16)	128	128x64
13	**Non-bt-1D** (dilated 2)	128	128x64
14	**Non-bt-1D** (dilated 4)	128	128x64
15	**Non-bt-1D** (dilated 8)	128	128x64
16	**Non-bt-1D** (dilated 16)	128	128x64
17	**Deconvolution** (upsampling)	64	256x128
18-19	2 x **Non-bt-1D**	64	256x128
20	**Deconvolution** (upsampling)	16	512x256
21-22	2 x **Non-bt-1D**	16	512x256
23	**Deconvolution** (upsampling)	C	1024x512

ENCODER

DECODER

Figure taken from paper [9]

ERF-Net: Architecture

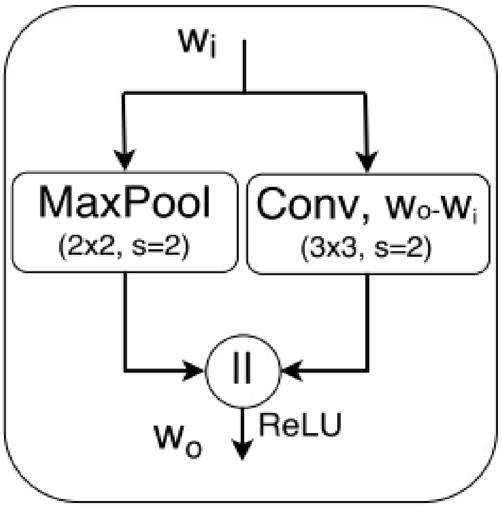


(a) Non-bottleneck-1D

```
(11): nn.Sequential {
    [input -> (1) -> (2) -> (3) -> output]
    (1): nn.ConcatTable {
      input
        |`-> (1): nn.Sequential {
        |       [input -> (1) -> (2) -> (3) -> (4) -> (5) -> (6) -> (7) -> (8) ->
(9) -> (10) -> output]
        |       (1): cudnn.SpatialConvolution(128 -> 128, 3x1, 1,1, 1,0)
        |       (2): cudnn.ReLU
        |       (3): cudnn.SpatialConvolution(128 -> 128, 1x3, 1,1, 0,1)
        |       (4): cudnn.SpatialBatchNormalization
        |       (5): cudnn.ReLU
        |       (6): nn.SpatialDilatedConvolution(128 -> 128, 3x1, 1,1, 8,0, 8,1)
        |       (7): cudnn.ReLU
        |       (8): nn.SpatialDilatedConvolution(128 -> 128, 1x3, 1,1, 0,8, 1,8)
        |       (9): cudnn.SpatialBatchNormalization
        |       (10): nn.SpatialDropout(0,3)
        |     }
         `-> (2): nn.Sequential {
                [input -> (1) -> output]
                (1): nn.Identity
              }
         ... -> output
    }
    (2): nn.CAddTable
    (3): cudnn.ReLU
```

Figure taken from paper [9]

ERF-Net: Architecture



(b) Downsampler block

Figure taken from paper [9]

```
(1): nn.Sequential {
  [input -> (1) -> (2) -> (3) -> (4) -> (5) -> output]
  (1): nn.ConcatTable {
    input
      |`-> (1): nn.Sequential {
      |       [input -> (1) -> output]
      |       (1): cudnn.SpatialConvolution(3 -> 13, 3x3, 2,2, 1,1)
      |     }
       `-> (2): nn.Sequential {
              [input -> (1) -> output]
              (1): cudnn.SpatialMaxPooling(2x2, 2,2)
            }
       ... -> output
  }
  (2): nn.JoinTable
  (3): cudnn.SpatialBatchNormalization
  (4): nn.SpatialDropout(0,000000)
  (5): cudnn.ReLU
```

ERF-Net performance on the Cityscapes dataset

name	fine	coarse	16-bit	depth	video	sub	IoU class	iIoU class	IoU category	iIoU category	Runtime [s]	code
m-TCFs	yes	yes	no	no	no	no	71.8	43.6	87.6	70.6	1.0	no
LRR-4x	yes	yes	no	no	no	no	71.8	47.9	88.4	73.9	n/a	yes
FRRN	yes	no	no	no	no	2	71.8	45.5	88.9	75.1	n/a	yes
Adelaide_context	yes	no	no	no	no	no	71.6	51.7	87.3	74.1	n/a	no
ML-CRNN	yes	no	no	no	no	no	71.2	47.1	87.7	72.5	n/a	no

...

Position 10/55 and 12/55

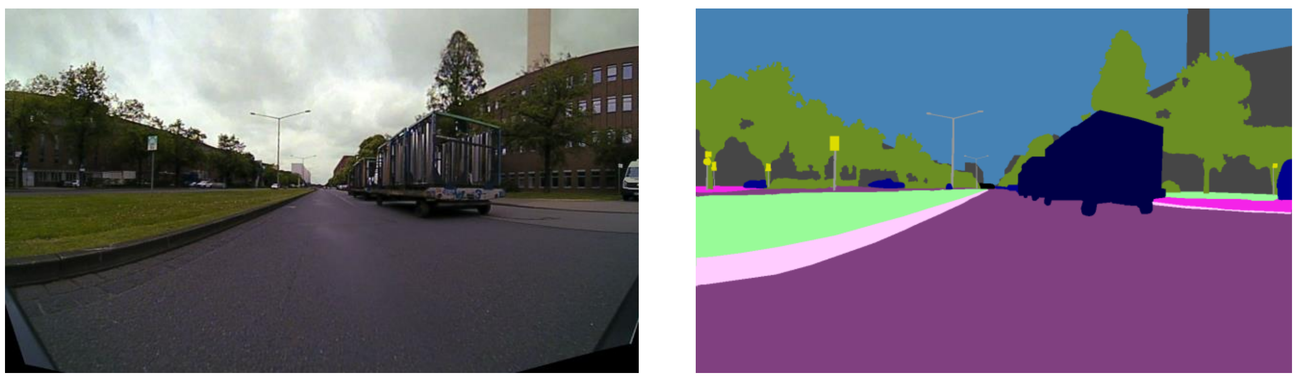
ERFNet (pretrained)	yes	no	no	no	no	2	69.7	44.1	87.3	72.7	0.02	yes
GridNet	yes	no	no	no	no	no	69.5	44.1	87.9	71.1	n/a	no
ERFNet (from scratch)	yes	no	no	no	no	2	68.0	40.4	86.5	70.4	0.02	yes

Transfer learning

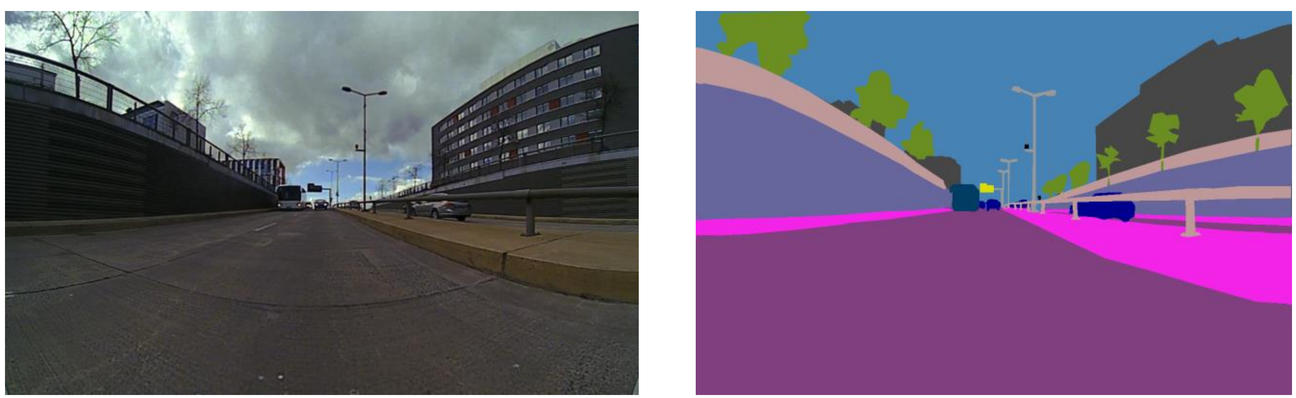
- Large annotated datasets are not always available for a given application
- Use **transfer learning** => train the network on a large existing dataset and use the weights to **fine-tune** the network for the specific task
- We perform different experiments in order to get the best results on our dataset (Up-drive)
- We fine-tune the ERF-Net network trained on the Cityscape dataset to fit our dataset

Transfer learning

- Our dataset contains 294 annotated images (front view, back view)
- 234 images in the training set
- 60 images in the validation set
- 20 semantic classes



Front view



Back view

Experimental results

- **Experiment 1**
  - Train encoder on the Cityscapes + Up-drive images
  - Train decoder on the Cityscapes + Up-drive images
  - Use data augmentation: image flipping and random translations => 234 * 2 = 468 Up-drive images (2975 Cityscapes + 468 Up-drive = 3443 images)
  - 8 classes: road, sidewalk, building, pole, vegetation, sky, person, vehicle
  - Training time: 60h for 300 epochs on 2 Nvidia GTX 1070
  - 84.71 IoU on Cityscapes validation set
  - 84.72 IoU on Up-drive validation set

Experimental results

	IoU Up-drive	IoU Cityscapes
CityScapes+ Up-driveAug	84.72	84.71
Pretrained enc ImageNet + Cityscapes+Up-driveAug decoder	86.7	85.87
Pretrained enc ImageNet + Up-driveAug decoder	91.14	55.53
Pretrained enc Cityscapes + Up-driveAug decoder	79.26	55.98
Up-driveAug	71.65	

Conclusions

- semantic segmentation can be done in real time
- best results on our dataset are obtained using a pretrained decoder on ImageNet and training the decoder on Cityscapes + Up-Drive images
- Future work: add more training and validation examples in our dataset and get a better classifier

Bibliography

[1] Everingham, Mark, et al. "The pascal visual object classes challenge: A retrospective." International journal of computer vision 111.1 (2015): 98-136.

[2] "The Role of Context for Object Detection and Semantic Segmentation in the Wild”, Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, Alan Yuille IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014

[3] "Microsoft coco: Common objects in context.”, Lin, Tsung-Yi, et al. European conference on computer vision. Springer, Cham, 2014.

[4] "Semantic understanding of scenes through the ADE20K dataset." Zhou, Bolei, et al.arXiv preprint arXiv:1608.05442 (2016).

[5] Assisted Video Object Labeling By Joint Tracking of Regions and Keypoints, Julien Fauqueur, Gabriel Brostow, Roberto Cipolla, IEEE International Conference on Computer Vision (ICCV'2007) Interactive Computer Vision Workshop. Rio de Janeiro, Brazil, October 2007

[6] The Cityscapes Dataset for Semantic Urban Scene Understanding,”, M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[7] „The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes,” G. Ros, L. Sellart, J. Materzynska, D. Vazquez and A. M. Lopez, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 3234-3243.

[8]Fully convolutional networks for semantic segmentation.”, Long, Jonathan, Evan Shelhamer, and Trevor Darrell. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[9] "Efficient ConvNet for Real-time Semantic Segmentation", E. Romera, J. M. Alvarez, L. M. Bergasa and R. Arroyo, IEEE Intelligent Vehicles Symposium (IV), pp. 1789-1794, Redondo Beach (California, USA), June 2017

Acknowledgment:
This work was supported by the EU H2020 project, UP-Drive under grant nr. 688652

Thank you!