Fully convolutional neural networks

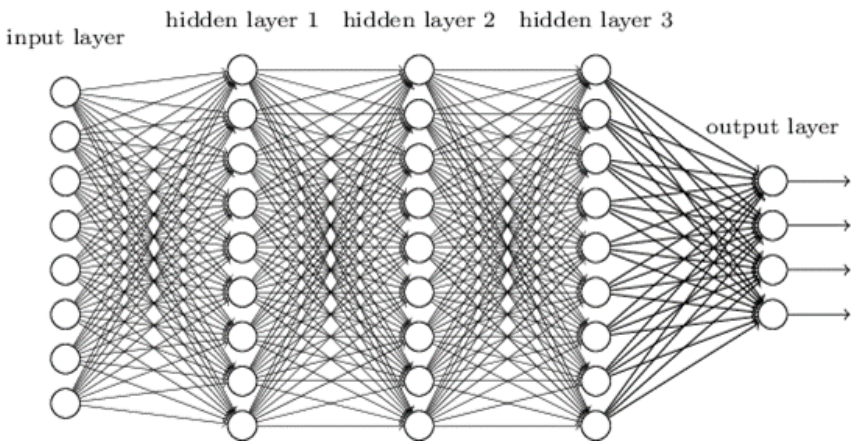
E. Decencière

MINES ParisTech
PSL Research University
Center for Mathematical Morphology

Contents

Contents

Recall from yesterday: image classification with NN



Simple fully-connected neural network

Cross-entropy loss function used for classification tasks:

$$L(\boldsymbol{\theta}) = -\sum_{i=1}^{n} y_i \log(f(\mathbf{x}_i, \theta))$$

Learning image transformations

- An image classification task is a function from the set of considered images into a set of labels
- In many applications, we want to transform an image into another image

Image definition

Definition: image

An 2-dimensional image I of size $p \times q$ $(p, q \in \mathbb{N}^*)$ is a function:

$$[0, \dots p-1] \times [0, \dots q-1] \longmapsto \mathbb{R}^d \quad (d \in \mathbb{N}^*)$$

The set of these images is $\mathcal{I}^d$.

Examples

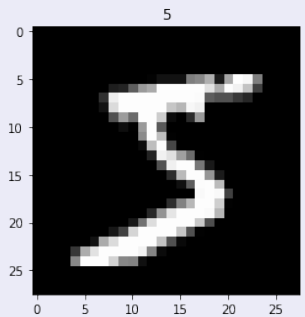


Figure: 28×28 grey level image $(d = 1)$ from the MNIST data set, and 481×321 colour image $(d = 3)$ from the Berkeley segmentation data set.

Image-to-image NN

Definition: image-to-image neural network

An image-to-image NN F is a NN that transforms an image into an image of same size[a]:

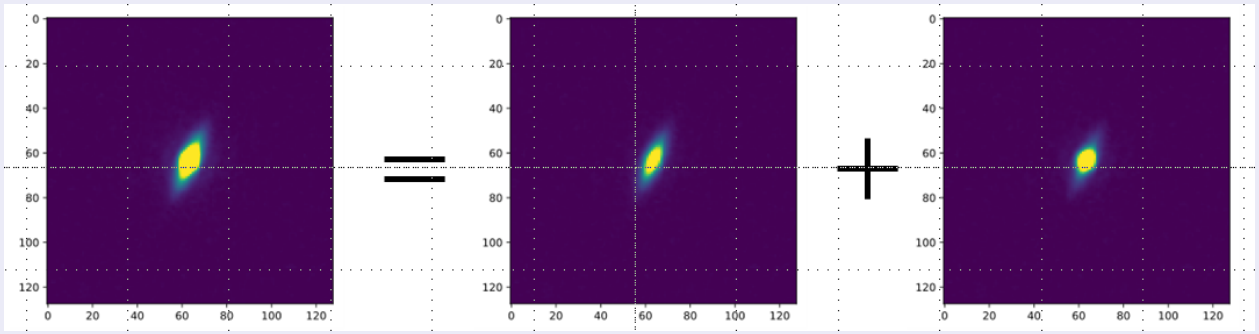
$$F : \mathcal{I}^{d_1} \longrightarrow \mathcal{I}^{d_2}$$
$$I \longmapsto F(I)$$

Note that the dimensions d_1 and d_2 of the value spaces can be different.

[a] In some applications the output size is different from the input size, but for the sake of simplicity we will not consider this case here

Examples

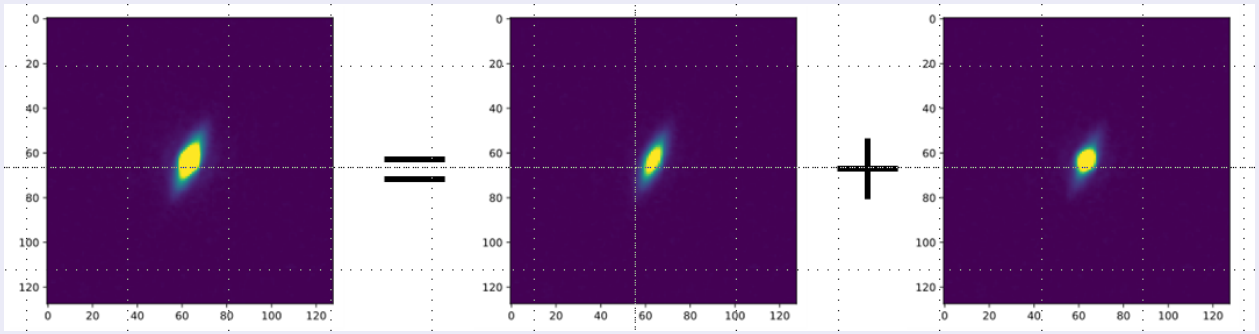
Bulge / disk decomposition



(Credits: Tuccillo, Huertas-Company, Velasco-Forero, Decencière)

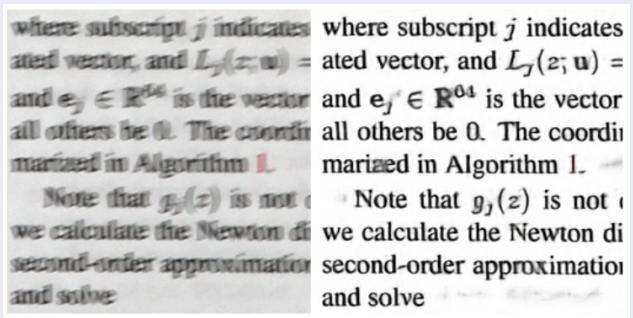
Examples

Bulge / disk decomposition



(Credits: Tuccillo, Huertas-Company, Velasco-Forero, Decencière)

Deblurring network [Hradiš et al., 2015]



where subscript j indicates
ated vector, and $L_j(z; \mathbf{u}) =$
and $\mathbf{e}_j \in \mathbb{R}^{64}$ is the vector
all others be 0. The coordi
marized in Algorithm 1.

Note that $g_j(z)$ is not
we calculate the Newton di
second-order approximatio
and solve

Image Segmentation with NNs

- Image segmentation often is an important step in an image processing work flow
- Image segmentation has been a very active deep learning research field

Image segmentation example

Credits: Pascal VOC database

Other applications

- Image filtering
- High dynamic range
- Style modification
- Super-resolution (image size increases)
- Motion estimation

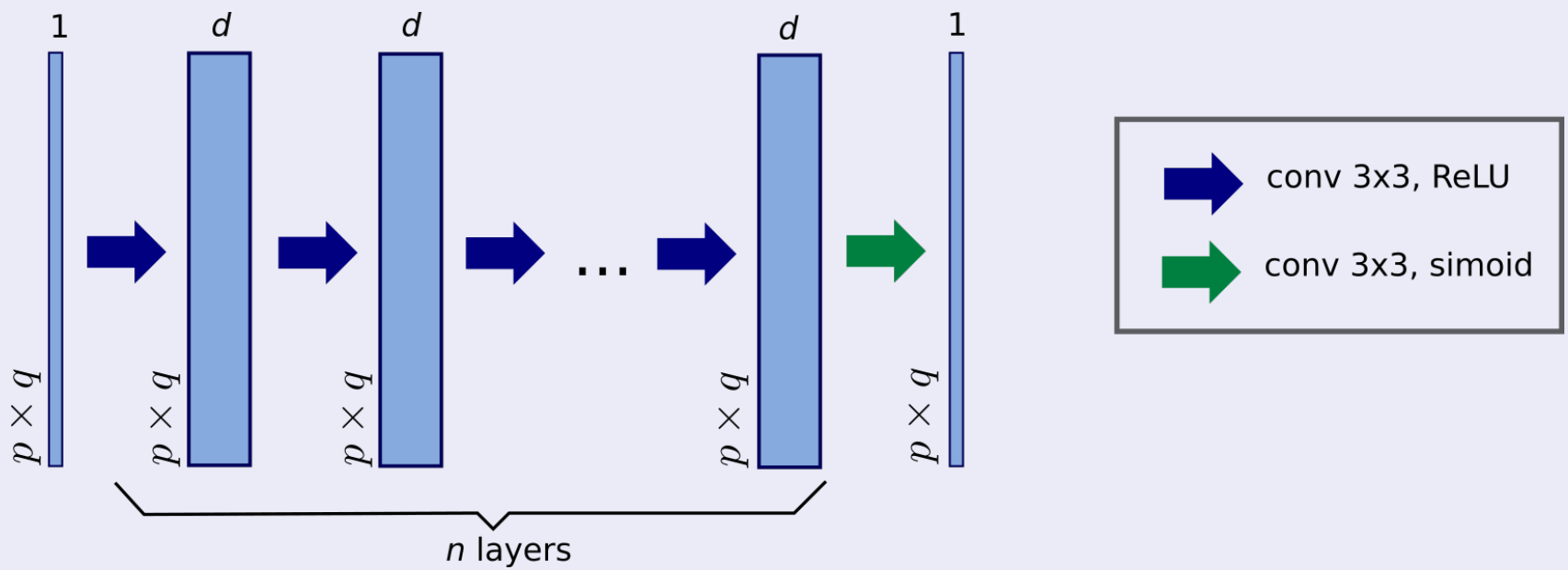
Image-to-image NNs architecture

- Image-to-image NNs are based on convolutional layers
- If downsampling is used, the corresponding upsampling is needed

Image-to-image NNs architecture

- Image-to-image NNs are based on convolutional layers
- If downsampling is used, the corresponding upsampling is needed

Example: plain CNN [Pang et al., 2010]



Receptive field

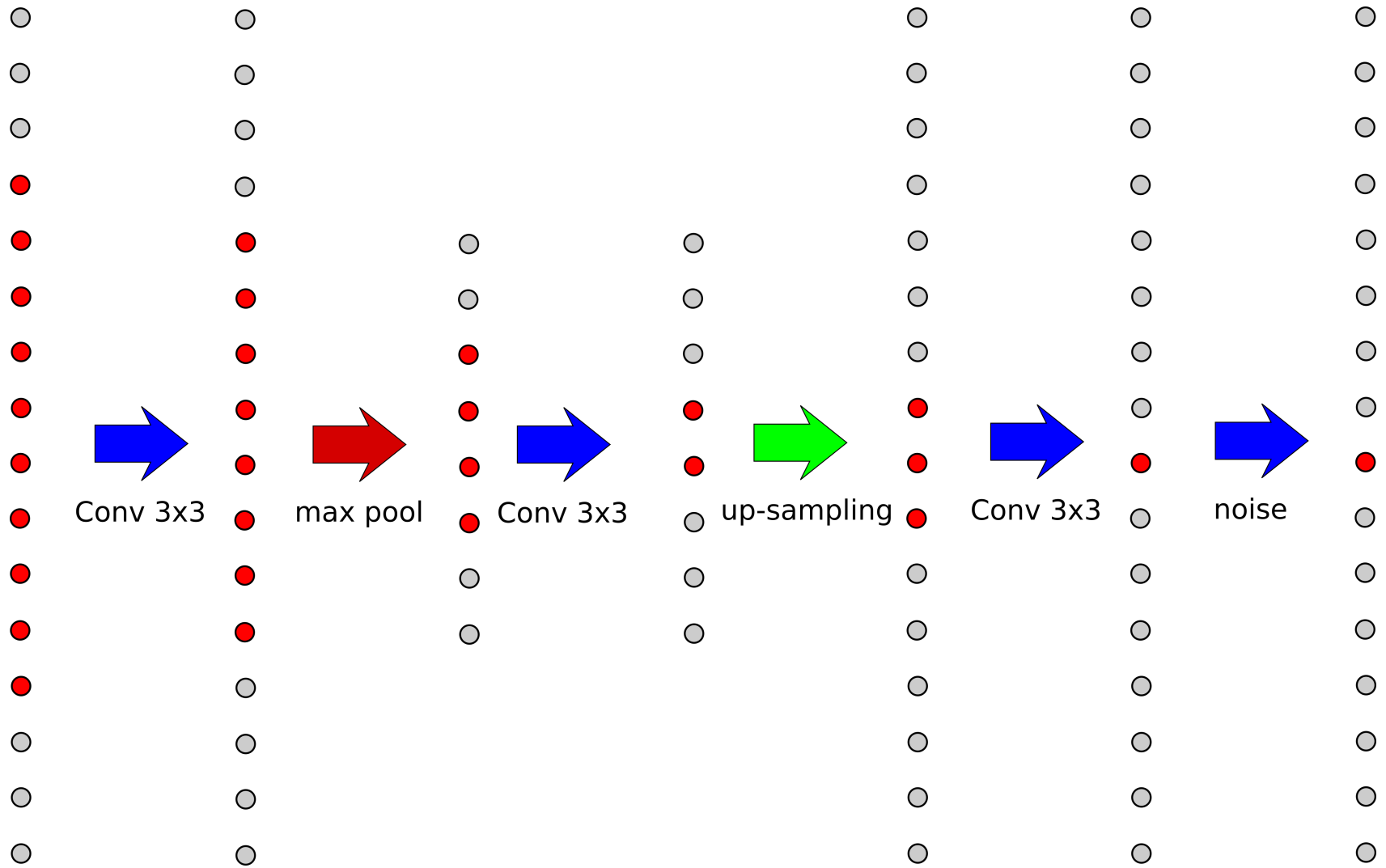
Definition: links between neurons

In a NN, we say that neuron a is linked to neuron b if there is an oriented path in the corresponding graph going from a to b.

Definition

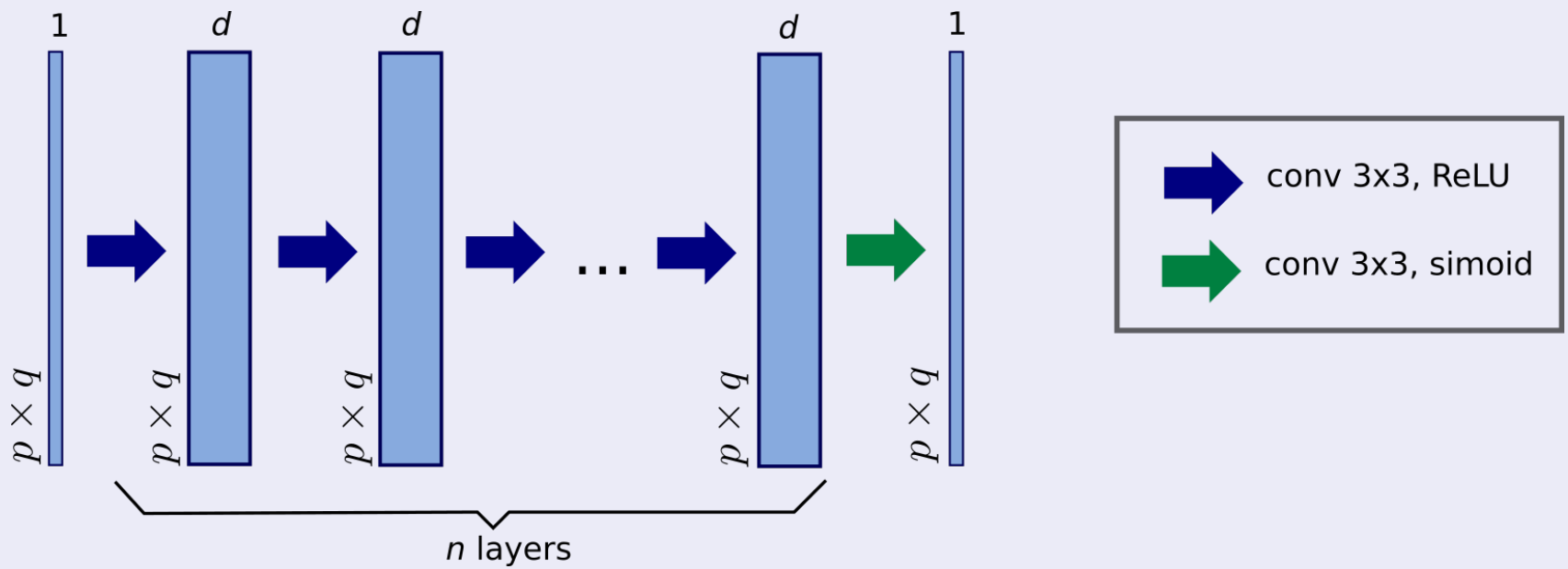
The receptive field of a neuron in a NN is the set of *input neurons* that are linked to that neuron.
The size of the receptive field is an essential property when designing a fully-convolutional NN architecture.

Illustration



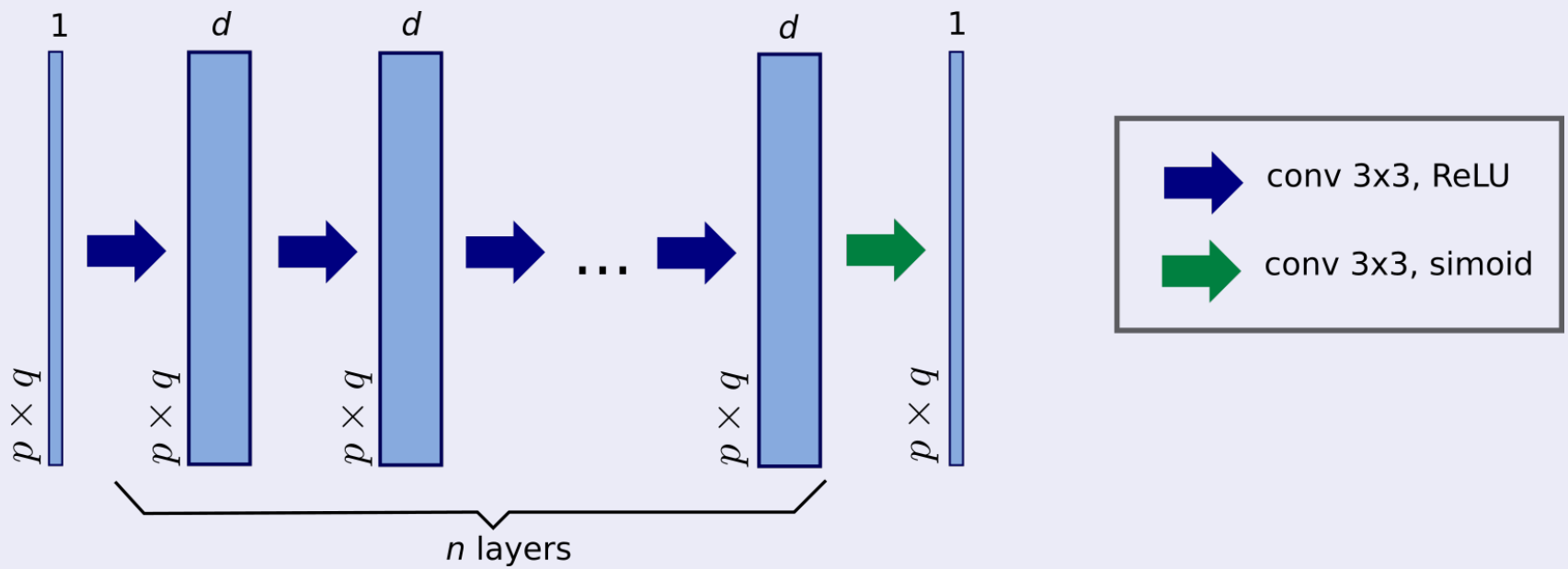
Example

What is the size of the receptive field of the neurons in the last layer?



Example

What is the size of the receptive field of the neurons in the last layer?



Answer: $1 + 2 \times (n + 1)$

The specific case of image segmentation

Definition: image segmentation

Let I be an image defined on D. A segmentation of I is a partition of D. In practice the regions of the segmentation should correspond to the objects in I, which is application dependant.

- A partition is often represented as a labelled image
- In order to make the segments symmetric, each one is represented by a different channel

Image segmentation example

Credits: Pascal VOC database

Some vocabulary on segmentation

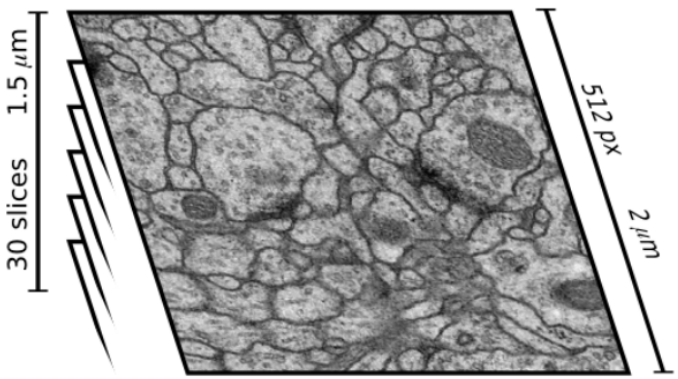
- **Object detection / localization**: bounding box around the object(s).
- **Binary segmentation**: segmentation in 2 classes, background and object.
- **Semantic segmentation**: a label is given to each pixel, according to the object it belongs to.
- **Instance segmentation**: identify each separate object, even if they belong to the same class.

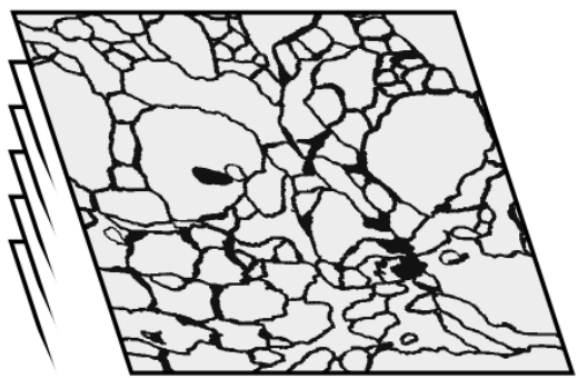
Contents

Contents

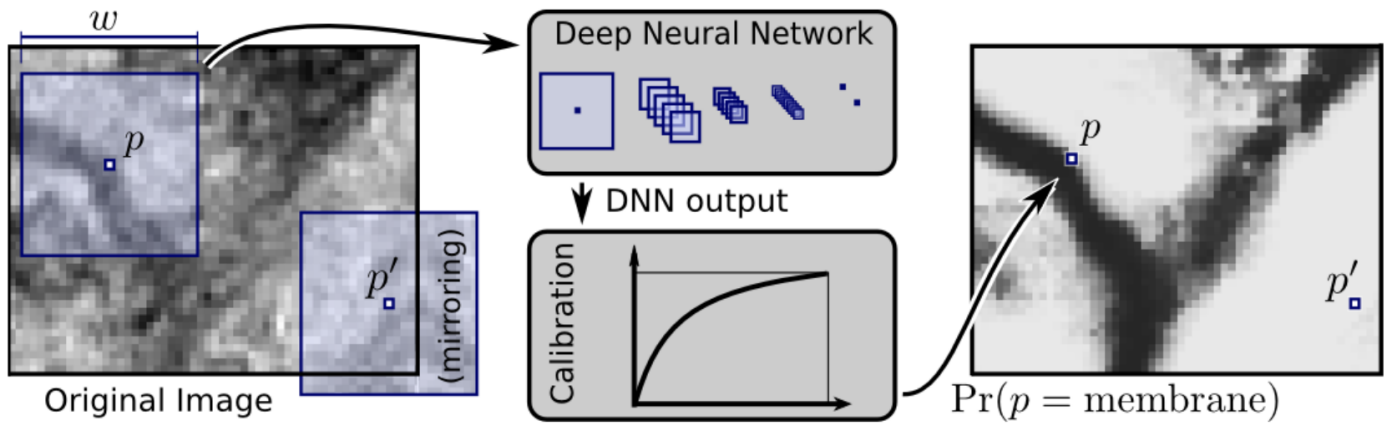
Neuron membrane segmentation challenge (ISBI 2012)

- Train: single stack of size $30 \times 512 \times 512$.
- Test: a second stack of same size.





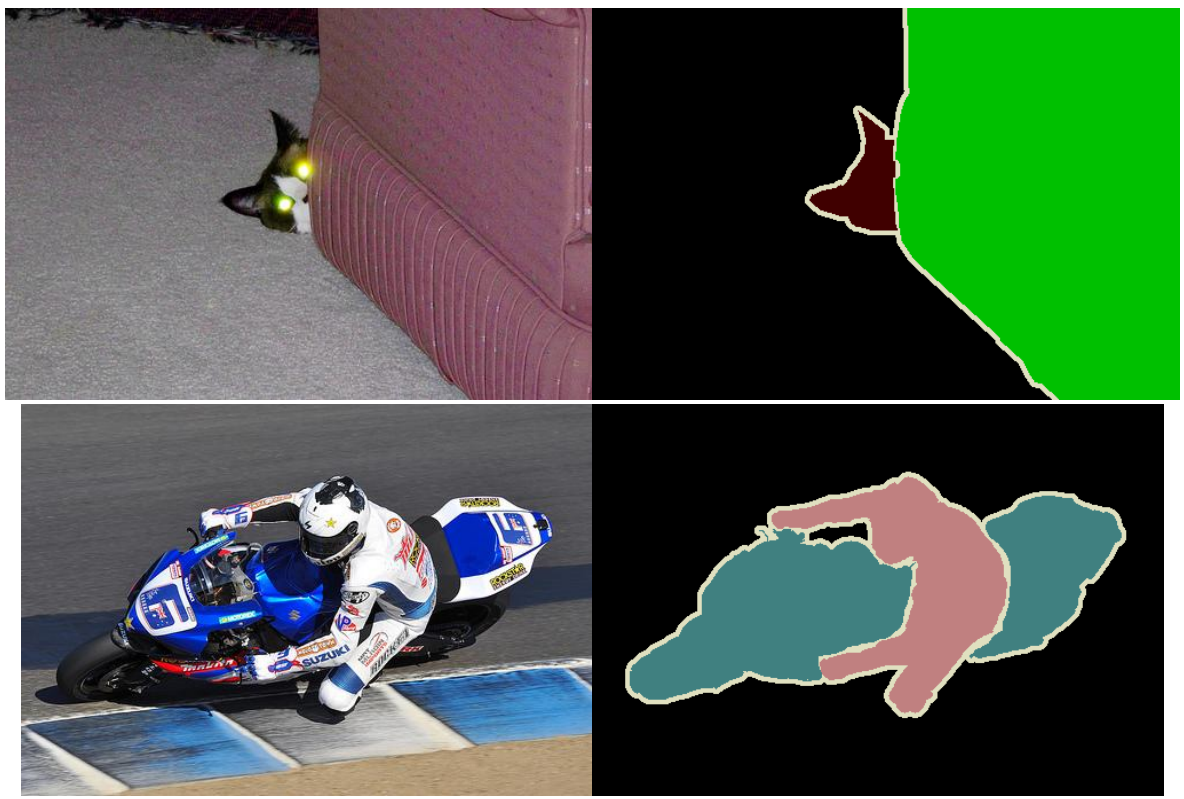
Neuron membrane segmentation challenge winner [Ciresan et al., 2012]



Contents

Pascal visual object classes segmentation challenge 2012 [Everingham et al., 2014]

- 1464 training and 1449 validation images
- automatic online test, with unknown images
- 20 image categories (cat, sofa, motorbike, person, etc.)

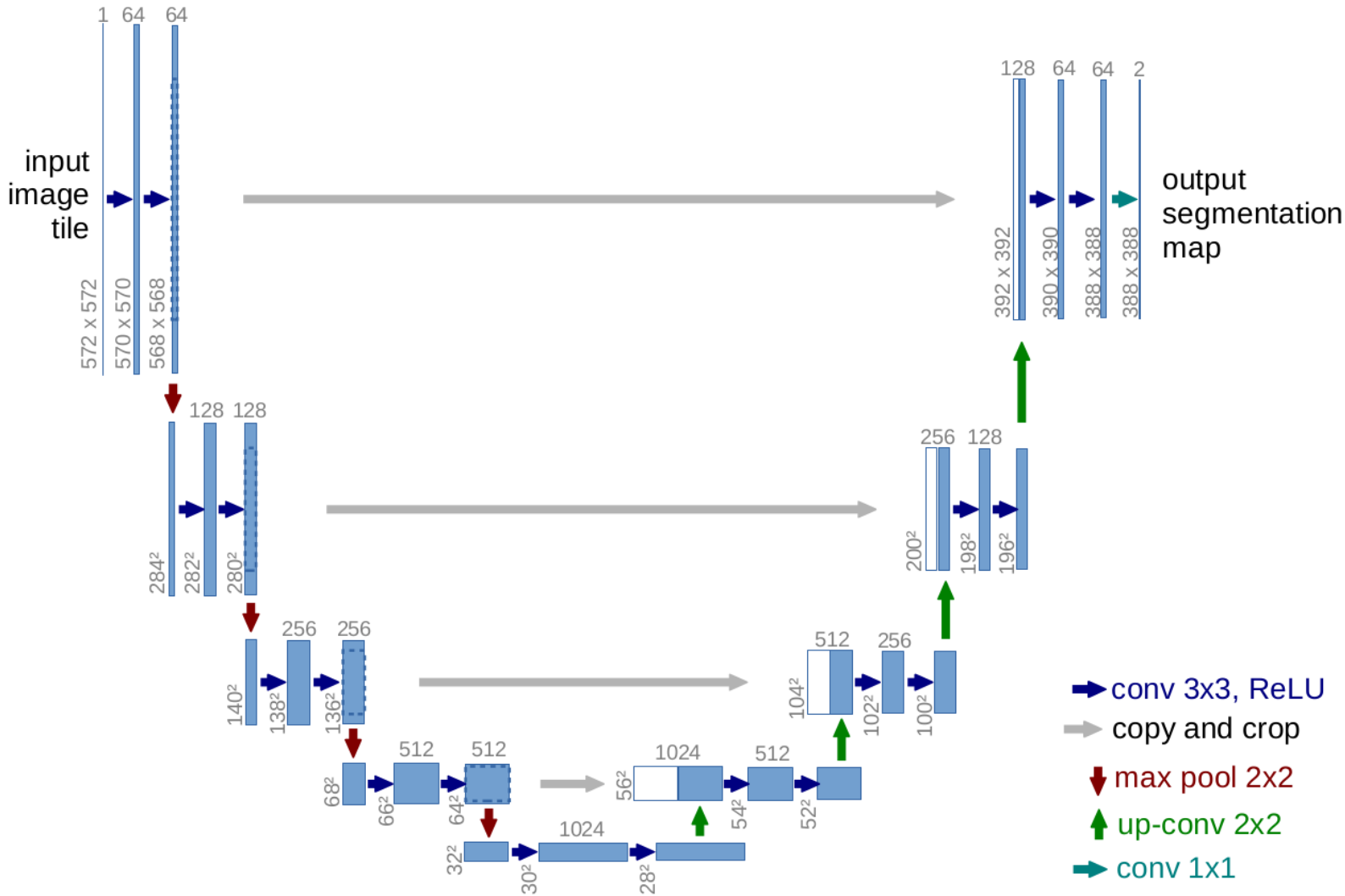


Convolutional nets for semantic image segmentation

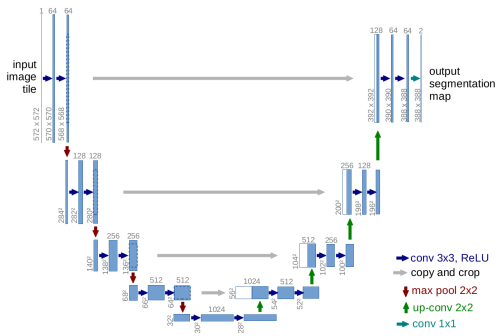
Three papers in 2015:

- Fully convolutional networks for semantic segmentation [Long et al., 2015]
- U-Net: convolutional networks for biomedical image segmentation [Ronneberger et al., 2015]
- SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation [Badrinarayanan et al., 2015]

Example: U-Net architecture [Ronneberger et al., 2015]



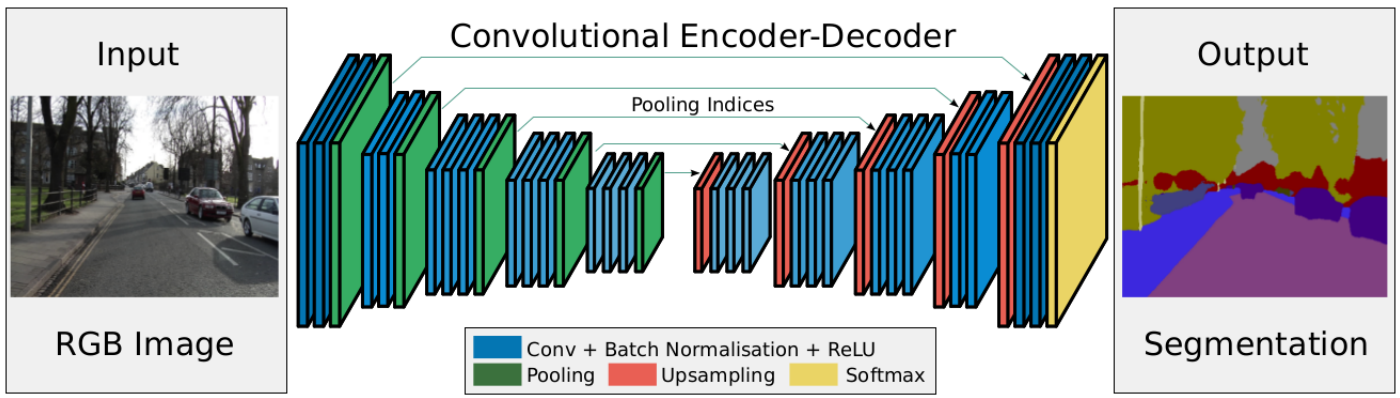
Notes on the U-Net architecture [Ronneberger et al., 2015]



- Activation of the last layer: soft-max
- Other activations: ReLU
- Loss used in the original publication: cross entropy with a weight map w to favor some pixels:

$$L(\boldsymbol{\theta}) = \sum_{M \in D} w(M) \log\big(\hat{y}_{l(M)}(M)\big)$$

Example: SegNet architecture [Badrinarayanan et al., 2015]



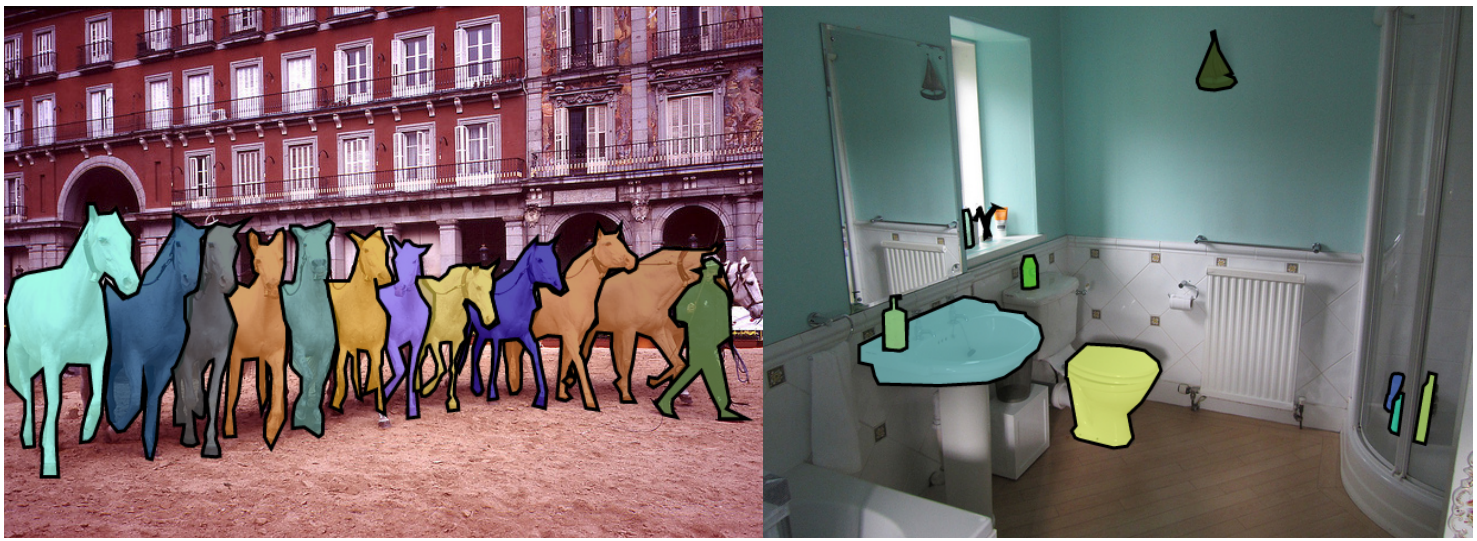
Remarks

- These architectures easily contain a number of parameters of the order of 10^7 (28 million for U-Net)
- Their optimization might be difficult
- For many segmentation applications, they are overkill
  - But you can reduce the number of filters or the number of layers

Contents

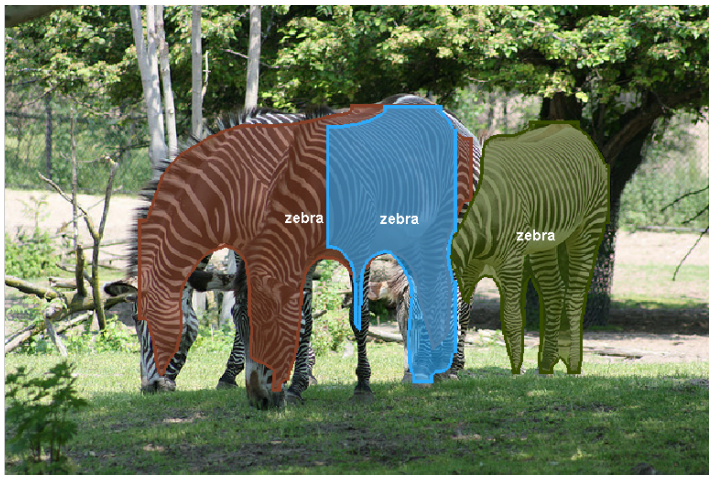
COCO: common objects in context [Lin et al., 2014]

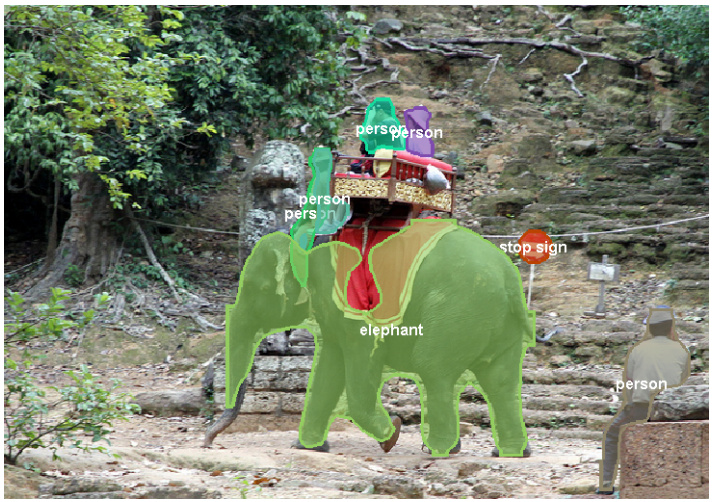
- 2 million objects, from 80 categories, in $300\,000$ images

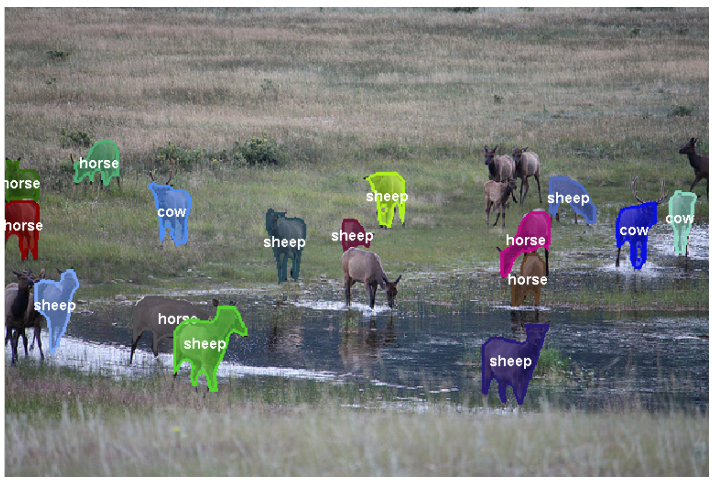


Winner 2016: Fully Convolutional Instance-aware Semantic Segmentation (Microsoft) [Li et al., 2016]

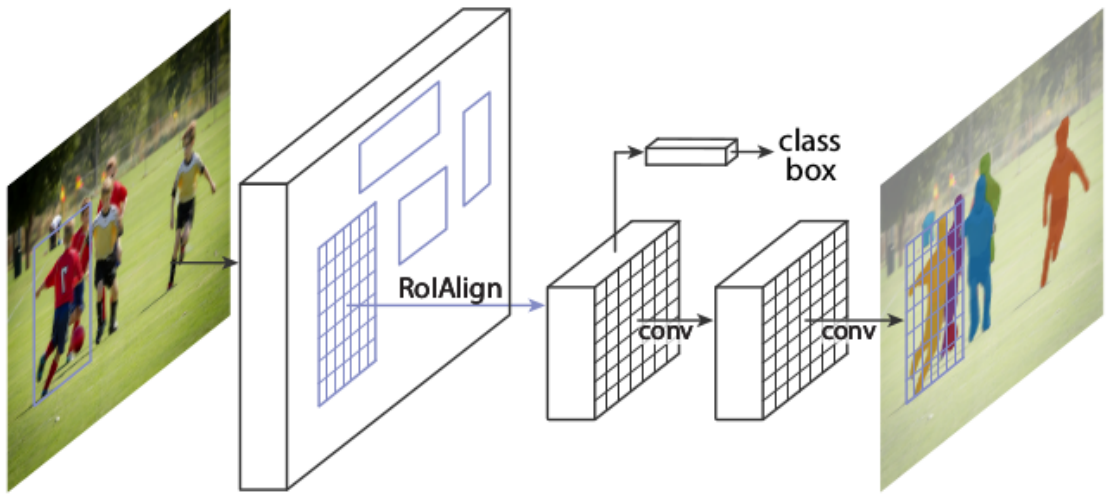
COCO instance segmentation challenge: examples of 2016 winner results



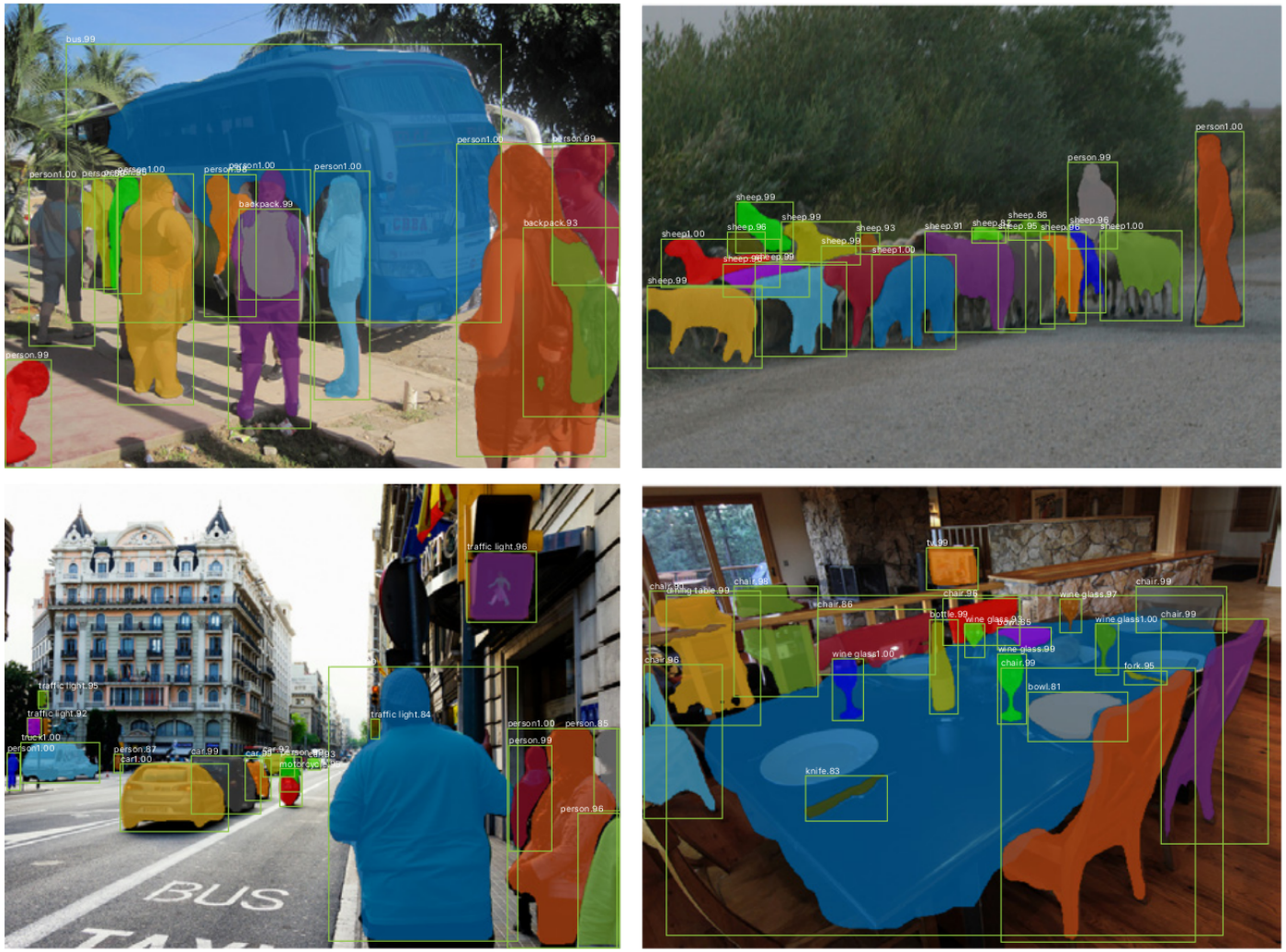




State of the art on the COCO database: Mask R-CNN [He et al., 2017]

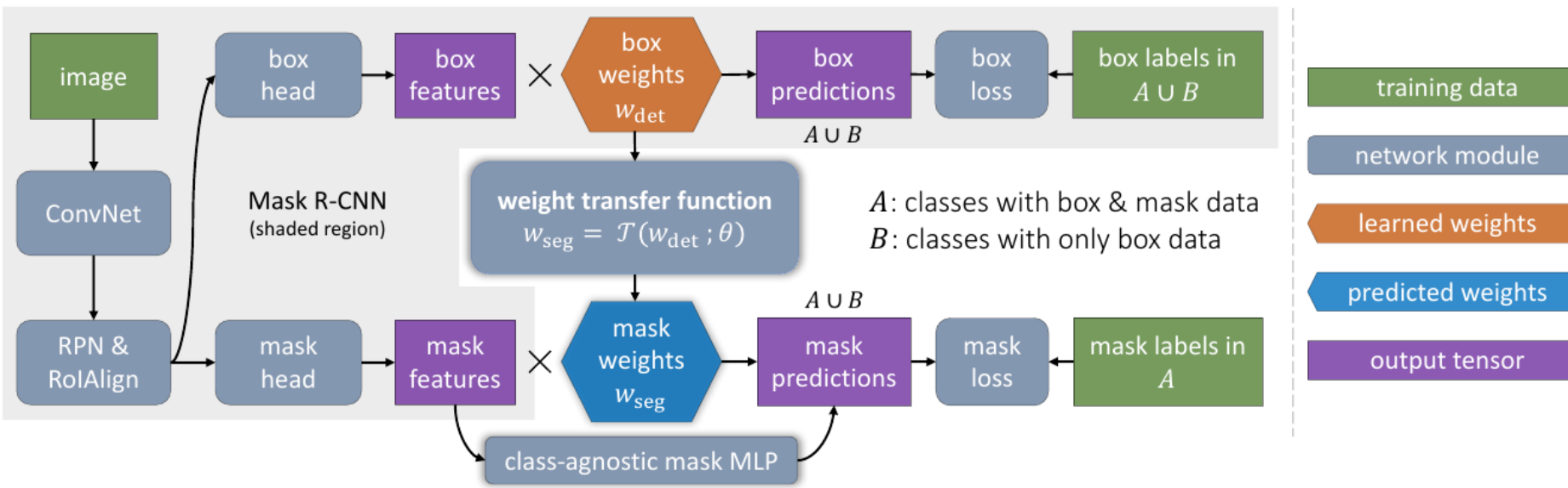


Mask R-CNN on the COCO database



Partially supervised segmentation - [Hu et al., 2017]

- 80 segmented categories from COCO database
- 3000 visual concepts using box annotations from the Visual Genome data set (100k images)



Current (?) trends for instance segmentation

- Region proposal +
- Fully convolutional (very deep) network +
- (Post-processing)

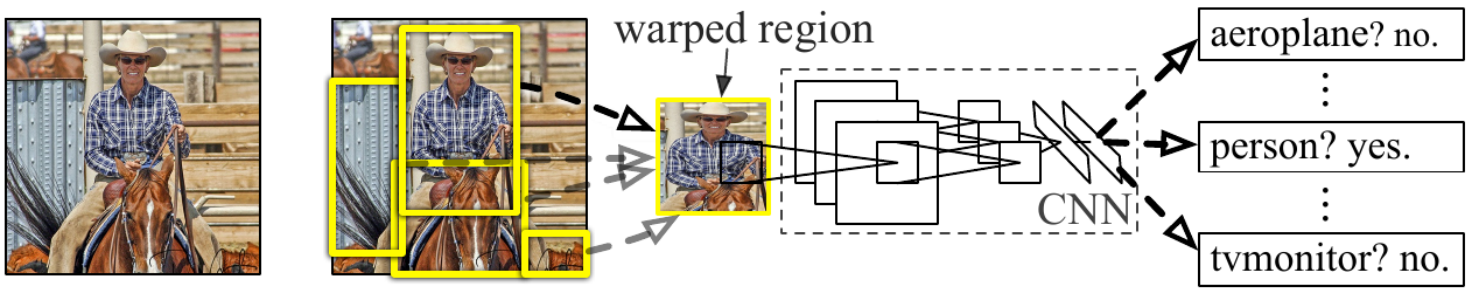


Figure: Regions with CNN features (R-CNN) (from [Girshick et al., 2014])

Current (?) trends for instance segmentation

- Region proposal +
- Fully convolutional (very deep) network +
- (Post-processing)

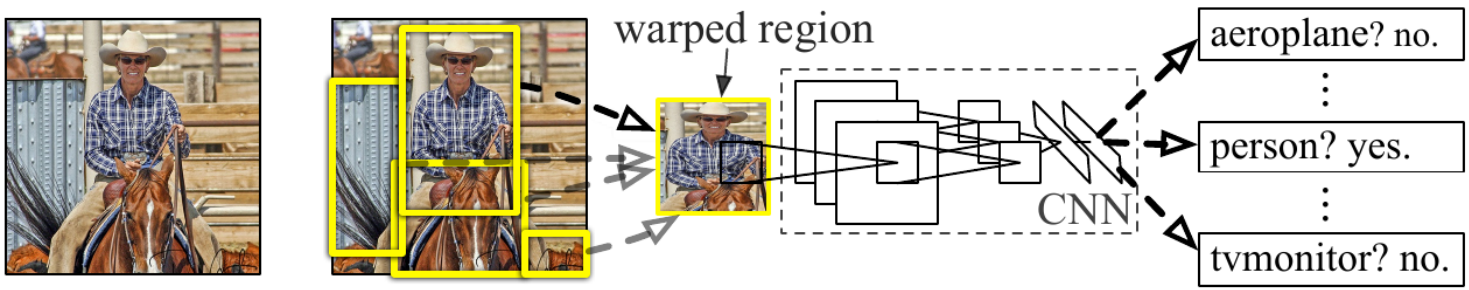


Figure: Regions with CNN features (R-CNN) (from [Girshick et al., 2014])

Meanwhile, on the object detection field...

- YOLO: you look only once [Redmon and Farhadi, 2016]
- SSD: single shot detector [Liu et al., 2016]

Contents

Translation invariance

Identical receptive fields produce identical outputs.

Translation invariance

Identical receptive fields produce identical outputs.

- If padding is used in the network, border effects can be important.

Translation invariance

Identical receptive fields produce identical outputs.

- If padding is used in the network, border effects can be important.
- Translation invariance is not always welcome!

Translation invariance

Identical receptive fields produce identical outputs.

- If padding is used in the network, border effects can be important.
- Translation invariance is not always welcome!
- Position information can also be used in the network:

Translation invariance

Identical receptive fields produce identical outputs.

- If padding is used in the network, border effects can be important.
- Translation invariance is not always welcome!
- Position information can also be used in the network:
  - Through masks or segmentations

Translation invariance

Identical receptive fields produce identical outputs.

- If padding is used in the network, border effects can be important.
- Translation invariance is not always welcome!
- Position information can also be used in the network:
  - Through masks or segmentations
  - Through pixel coordinates

Image size flexibility

- A NN containing fully-connected layers can only process images of a given size
- A translation invariant NN can be applied to images of any size, as long as its dimensions are compatible with the subsampling steps of the network
- Practical limit: the memory of the system

Image size flexibility

- A NN containing fully-connected layers can only process images of a given size
- A translation invariant NN can be applied to images of any size, as long as its dimensions are compatible with the subsampling steps of the network
- Practical limit: the memory of the system
- Note that as the input image gets larger, border effects become proportionally less present

Robustness with respect to ground-truth errors

This is more an empirical observation than a mathematical property, but fully-convolutional NNs tend to be robust with respect to errors in the contours position on the ground-truth.

Contents

Dealing with image sizes during training

- In segmentation applications, original images are often of different sizes and possibly very large.

Dealing with image sizes during training

- In segmentation applications, original images are often of different sizes and possibly very large.
- In theory, given the translation invariance of fully-convolutional NN, we could use them directly as input. In practice, we are limited by memory size.

Dealing with image sizes during training

- In segmentation applications, original images are often of different sizes and possibly very large.
- In theory, given the translation invariance of fully-convolutional NN, we could use them directly as input. In practice, we are limited by memory size.
- Solution: extract fixed-sized crops from your training set:

Dealing with image sizes during training

- In segmentation applications, original images are often of different sizes and possibly very large.
- In theory, given the translation invariance of fully-convolutional NN, we could use them directly as input. In practice, we are limited by memory size.
- Solution: extract fixed-sized crops from your training set:
  - make them as large as possible, to reduce border effects

Dealing with image sizes during training

- In segmentation applications, original images are often of different sizes and possibly very large.
- In theory, given the translation invariance of fully-convolutional NN, we could use them directly as input. In practice, we are limited by memory size.
- Solution: extract fixed-sized crops from your training set:
  - make them as large as possible, to reduce border effects
  - take a small batch size (1, 2, 4?)

Post-processing for segmentation

- Superpixels (e.g. [Farabet et al., 2013])
- Conditional random fields
- Mathematical morphology

Loss functions for image segmentation

- $\hat{\mathbf{y}} = (\hat{\mathbf{y}}_i)$: network output
- $\mathbf{y} = (\mathbf{y}_i)$: binary expected output
- We suppose that all $\hat{\mathbf{y}}_i$ are in $[0, 1]$
- We want the $\hat{\mathbf{y}}$ to be *as close as possible* to $\mathbf{y}$

Loss functions for image segmentation

Most commonly used loss functions

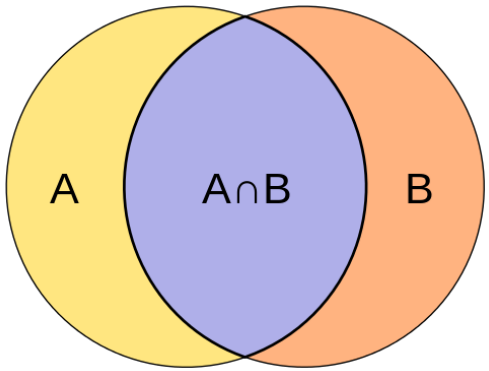
- Mean squared error (MSE): $\sum_i (\hat{y}_i - y_i)^2$
- Cross-entropy: $-\sum_i y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)$

Measures used in image processing

Let A and B be two sets, not simultaneously empty.

Dice coefficient

$$D(A,B) = \frac{2|A \cap B|}{|A| + |B|}$$



Jaccard index

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Properties

- $\forall A, B : 0 \leq J(A,B) \leq D(A,B) \leq 1$
- If $A = B$, then $D(A,B) = J(A,B) = 1$
- If $A \cap B = \emptyset$, then $D(A,B) = J(A,B) = 0$

Credits: Public domain (Wikipedia.org)

Generalization to $[0, 1]$

$\mathbf{y}$ and $\hat{\mathbf{y}}$ are in $[0, 1]^n$, not simultaneously equal to 0.

Dice similarity

$$D(\mathbf{y}, \hat{\mathbf{y}}) = \frac{2 \sum_i y_i \hat{y}_i}{\sum_i y_i + \sum_i \hat{y}_i}$$

Jaccard similarity

$$J(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\sum_i y_i \hat{y}_i}{\sum_i y_i + \sum_i \hat{y}_i - \sum_i y_i \hat{y}_i}$$

Corresponding loss functions

$\mathbf{y}$ and $\hat{\mathbf{y}}$ are in $[0, 1]^n$, not simultaneously equal to 0.

Dice loss

$$d(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{2 \sum_i y_i \hat{y}_i}{\sum_i y_i + \sum_i \hat{y}_i}$$

Jaccard loss

$$j(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{\sum_i y_i \hat{y}_i}{\sum_i y_i + \sum_i \hat{y}_i - \sum_i y_i \hat{y}_i}$$

In practice, these two losses give similar results.

Corresponding loss functions - variants

$\mathbf{y}$ and $\hat{\mathbf{y}}$ are in $[0, 1]^n$, not simultaneously equal to 0.
Constant ϵ, which is typically "small", keeps the denominator "far enough" from zero.

Dice loss

$$d(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{2 \sum_i y_i \hat{y}_i}{\sum_i y_i^2 + \sum_i \hat{y}_i^2 + \epsilon}$$

Jaccard loss

$$j(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{\sum_i y_i \hat{y}_i}{\sum_i y_i^2 + \sum_i \hat{y}_i^2 - \sum_i y_i \hat{y}_i + \epsilon}$$

These variants seem to work similarly to the original version. To the extent of my knowledge, there have been no studies on their respective merits.

Conclusion on loss functions

- Use the Jaccard loss as base line for segmentation problems
- Note that these losses compute their values pixel-wise: they do not take into account any structure (for example, continuity)
- Working on specific losses enforcing structure might be an interesting research path...

Contents

Image segmentation: a solved problem?

- Progress in image segmentation since 2012 has been enormous
- Several complex problems have now satisfactory solutions
- Training can be a problem (large annotated databases, difficult optimization)
- There are still challenges ahead...

Some research subjects

- Optimization - a very general, and essential, subject
- Making training databases as small as possible
- Specific losses
- Taking *a priori* structural information into account

References I

[Badrinarayanan et al., 2015] Badrinarayanan, V., Kendall, A., and Cipolla, R. (2015). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *arXiv:1511.00561 [cs]*. arXiv: 1511.00561.

[Ciresan et al., 2012] Ciresan, D., Giusti, A., Gambardella, L. M., and Schmidhuber, J. (2012). Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 2843–2851. Curran Associates, Inc.

[Everingham et al., 2014] Everingham, M., Eslami, S. M. A., Gool, L. V., Williams, C. K. I., Winn, J., and Zisserman, A. (2014). The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision*, 111(1):98–136.

[Farabet et al., 2013] Farabet, C., Couprie, C., Najman, L., and LeCun, Y. (2013). Learning Hierarchical Features for Scene Labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929.

[Girshick et al., 2014] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *Proceedings CVPR*, pages 580–587.

[He et al., 2017] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask R-CNN. *arXiv:1703.06870 [cs]*. arXiv: 1703.06870.

References II

[Hradiš et al., 2015] Hradiš, M., Kotera, J., Zemcík, P., and Šroubek, F. (2015). Convolutional neural networks for direct text deblurring. In *Proceedings of BMVC*, volume 10.

[Hu et al., 2017] Hu, R., Dollár, P., He, K., Darrell, T., and Girshick, R. (2017). Learning to Segment Every Thing. *arXiv:1711.10370 [cs]*. arXiv: 1711.10370.

[Li et al., 2016] Li, Y., Qi, H., Dai, J., Ji, X., and Wei, Y. (2016). Fully Convolutional Instance-aware Semantic Segmentation. *arXiv:1611.07709 [cs]*. arXiv: 1611.07709.

[Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2014). Microsoft COCO: Common Objects in Context. *arXiv:1405.0312 [cs]*. arXiv: 1405.0312.

[Liu et al., 2016] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. *arXiv:1512.02325 [cs]*, 9905:21–37. arXiv: 1512.02325.

[Long et al., 2015] Long, J., Shelhamer, E., and Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440.

References III

[Pang et al., 2010] Pang, B., Zhang, Y., Chen, Q., Gao, Z., Peng, Q., and You, X. (2010). Cell Nucleus Segmentation in Color Histopathological Imagery Using Convolutional Networks. In *2010 Chinese Conference on Pattern Recognition (CCPR)*, pages 1–5.

[Redmon and Farhadi, 2016] Redmon, J. and Farhadi, A. (2016). YOLO9000: Better, Faster, Stronger. *arXiv:1612.08242 [cs]*. arXiv: 1612.08242.

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, number 9351 in Lecture Notes in Computer Science, pages 234–241. Springer International Publishing.