Fully convolutional neural networks

E. Decencière

MINES ParisTech
PSL Research University
Center for Mathematical Morphology

Contents

Contents

Image definition

Definition

Here an image is a 2D array of size $p \times q$. Each array element belongs to $\mathbb{R}^d$. The dimension of the value space d, is often called the number of channels of the image.
The set of these images is noted $\mathcal{I}^d$.

Examples

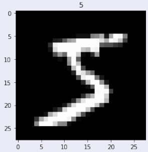
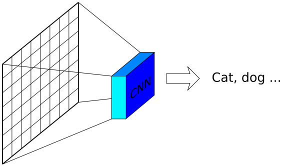


Figure: 28×28 grey level image $(d = 1)$ from the MNIST data set, and 481×321 colour image $(d = 3)$ from the Berkeley segmentation data set.

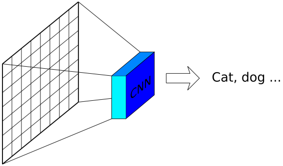
Learning image transformations

- An image classification task is a function from the set of considered images into a set of labels



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

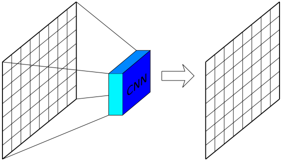


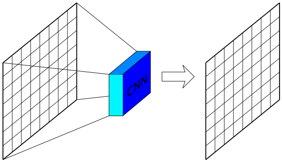
Image-to-image translation

Definition: image-to-image translation

An image-to-image operator is a function that transforms an image into another image of same size:

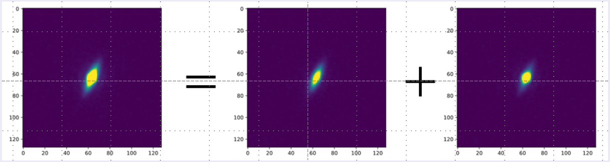
$$F : \mathcal{I}^{d_1} \longrightarrow \mathcal{I}^{d_2}$$
$$I \longmapsto J$$

Note that the number of channels of input and output images can be different.



Examples

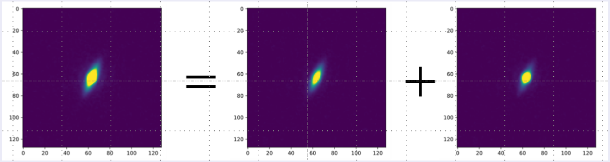
Bulge / disk decomposition



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

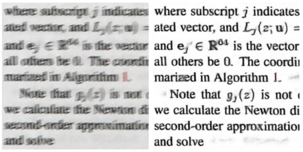
Examples

Bulge / disk decomposition

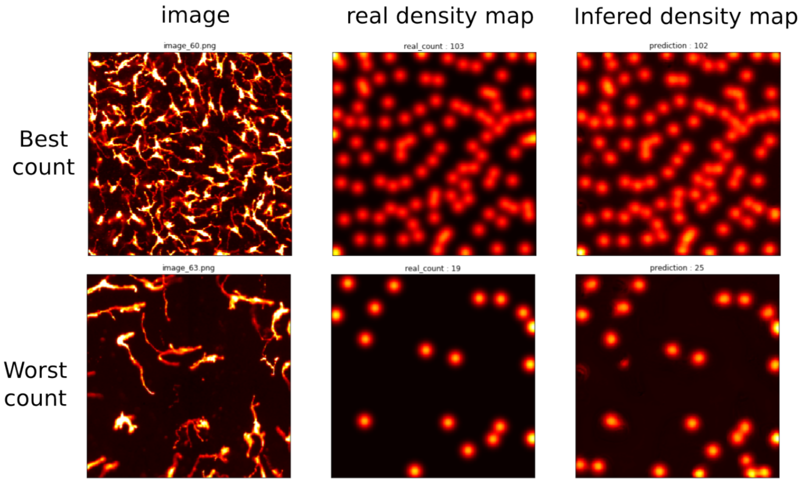


(Credits: Tuccillo, Huertas-Company, Velasco-Forero, Decencière)

Deblurring network [Hradiš et al., 2015]



Counting cells



Credits: Tristan Lazard, master thesis.
Images: L'Oréal.

Microscopy cross-modality prediction[Ounkomol et al., 2018]

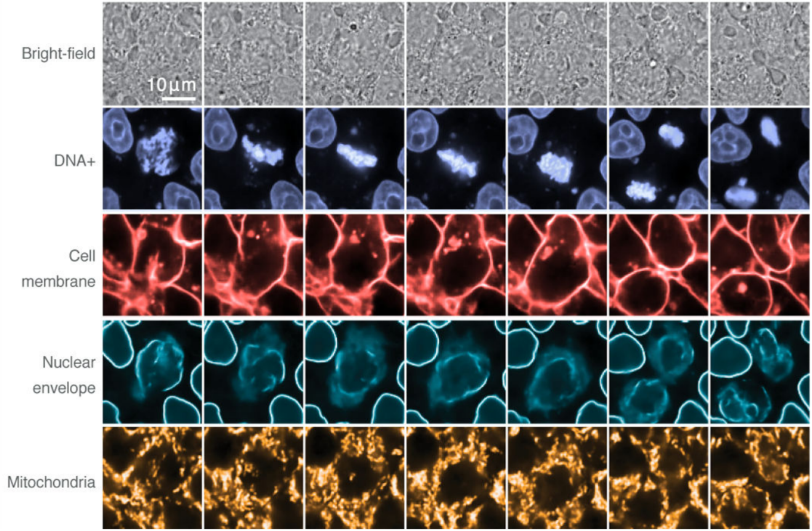
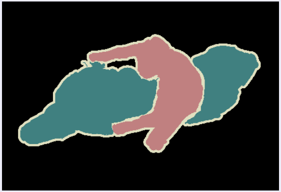


Image segmentation

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

Example



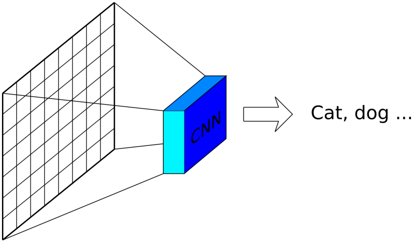
Credits: Pascal VOC database

Other applications

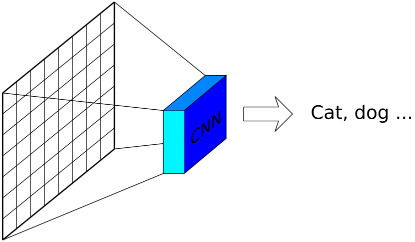
- Image filtering
- Style modification
- Motion estimation
- etc ...

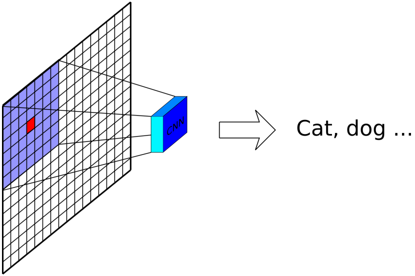
Contents

From classification nets to image-to-image nets

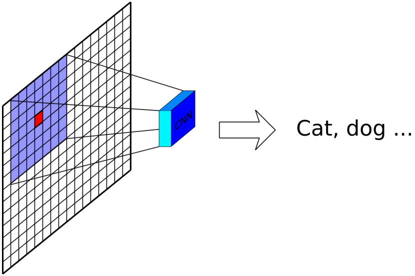


From classification nets to image-to-image nets

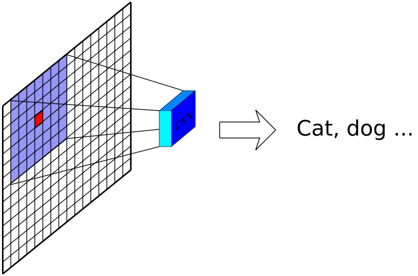


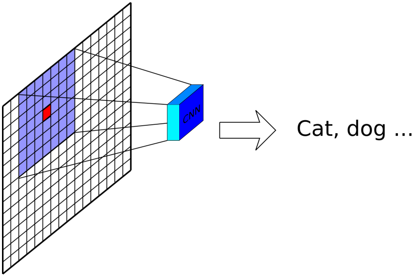


From classification nets to image-to-image nets

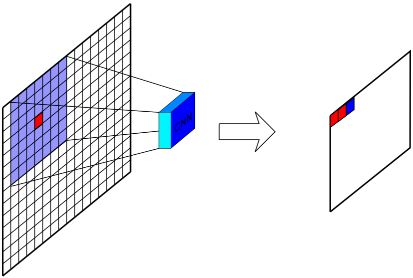


From classification nets to image-to-image nets

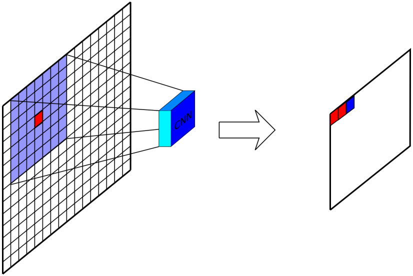




From classification nets to image-to-image nets



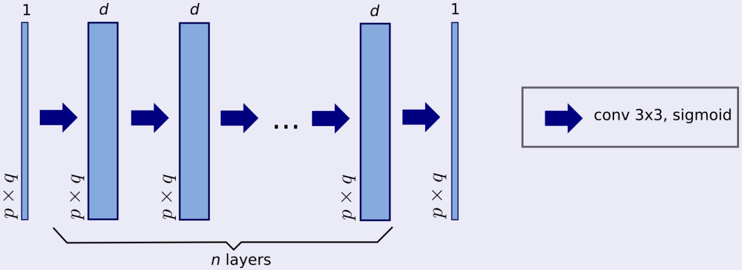
From classification nets to image-to-image nets



- The neuron membrane segmentation challenge winner [Cireșan et al., 2012] used this strategy. It is inefficient.
- The idea to compute the whole output image with a single pass through the network had already been proposed [Feng Ning et al., 2005, Jain et al., 2007] (but with architectures more complex than those used today).

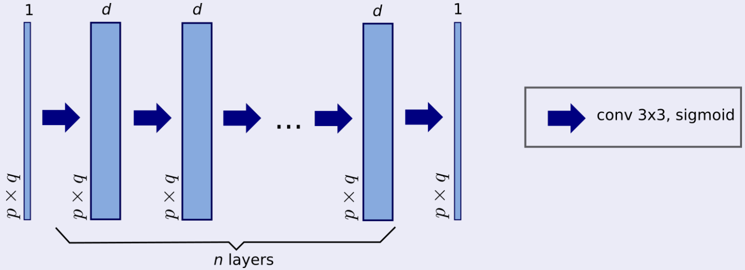
The simplest image-to-image architecture

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



The simplest image-to-image architecture

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

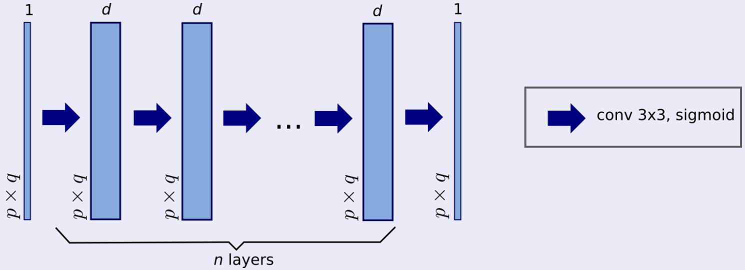


Note that today we would use ReLU activations on all layers except the last.

How many parameters does the first layer contain? The last? The second?

The simplest image-to-image architecture

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

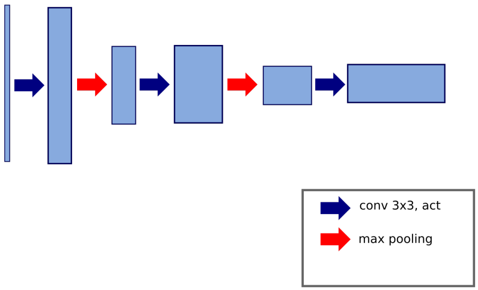


Note that today we would use ReLU activations on all layers except the last.

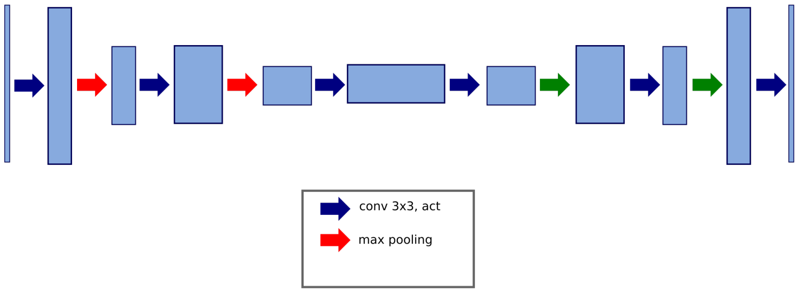
How many parameters does the first layer contain? The last? The second?

$(9+1) \times d$; $9d+1$; $(9d+1) \times d$

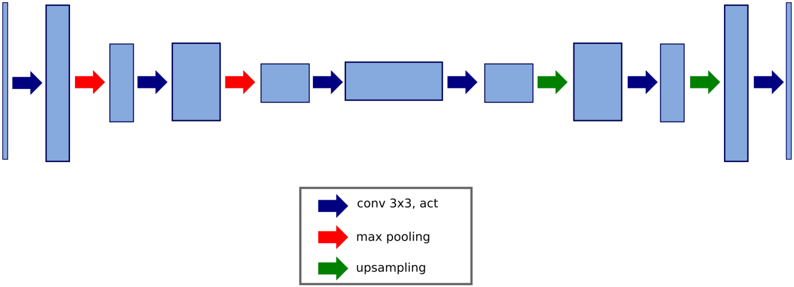
Going back to the original image size



Going back to the original image size



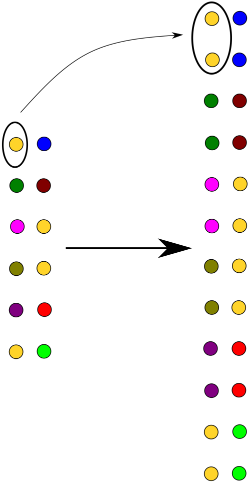
Going back to the original image size



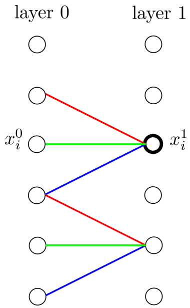
Upsampling techniques

- Replication
- Transposed convolution
- Pooling index memorization

Upsampling through replication

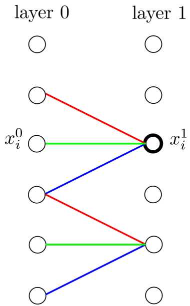


Transposed convolution

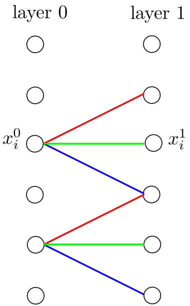


Convolutional layer

Transposed convolution

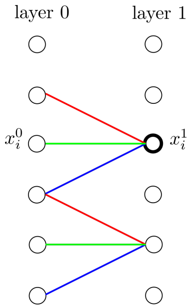


Convolutional layer

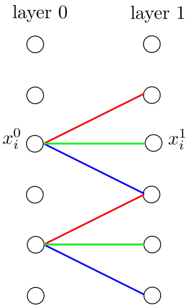


Transposed convolutional layer

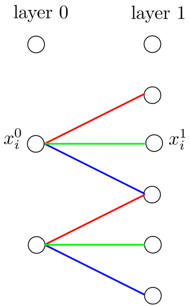
Transposed convolution



Convolutional layer



Transposed convolutional layer



Transposed convolutional layer with stride $= 2$

Contents

Contents

Receptive field

Definition: links between neurons

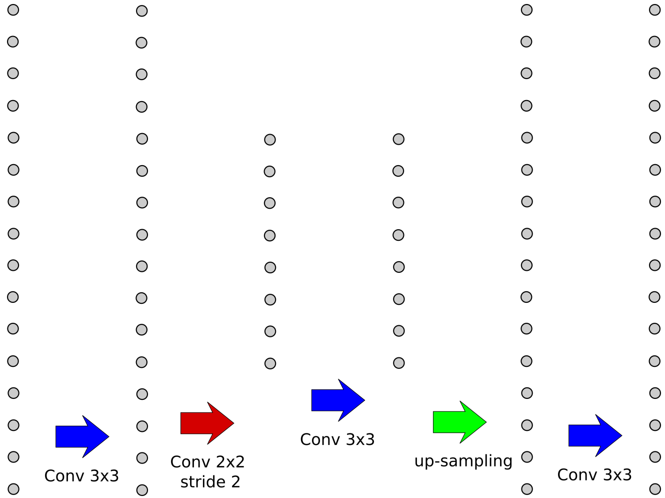
In a NN, we say that neuron a is linked to neuron b if there is an oriented path in the corresponding graph going from a to b.

Definition

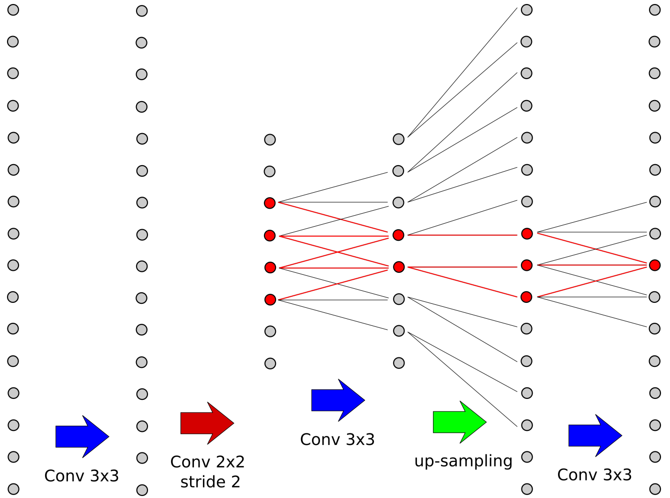
The receptive field of a neuron in a NN is the set of *input neurons* that are linked to that neuron.
The size of the receptive field is an essential property when designing a fully-convolutional NN architecture.

Note that in most cases, receptive fields are square shaped (border effects ignored).

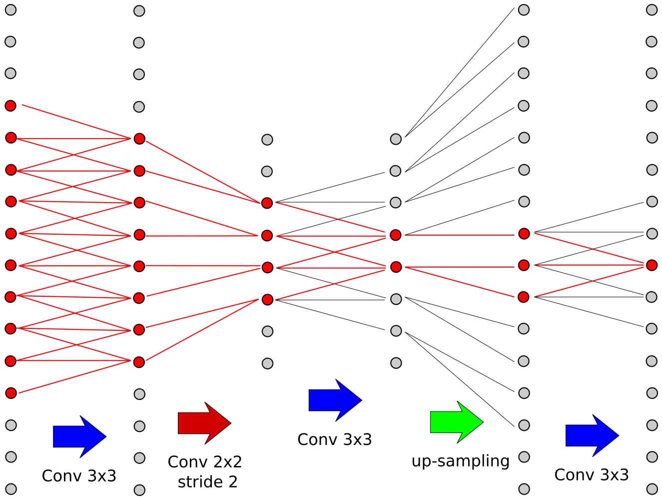
Illustration



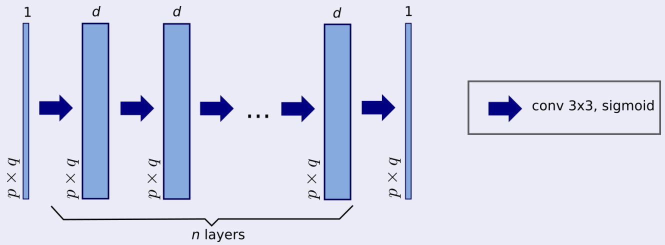
Illustration



Illustration



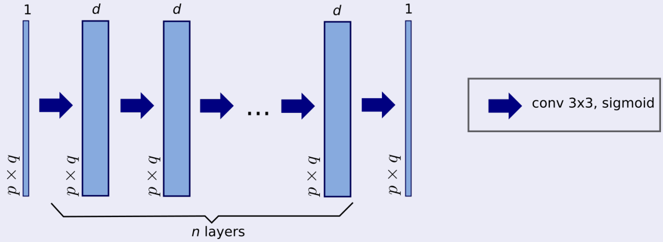
Example



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

1. n
2. $1 + 2 \times n$
3. $1 + 2 \times (n + 1)$
4. It depends on the input image

Example

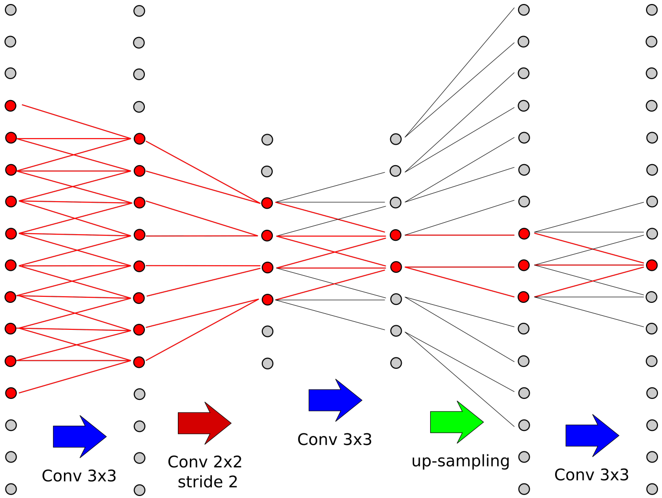


What is the width of the receptive field of the neurons in the last layer?

1. n
2. $1 + 2 \times n$
3. $1 + 2 \times (n + 1)$
4. It depends on the input image

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

Effective receptive field: intuition



Criterion

For a given output neuron y, what criterion seems sound to estimate the impact of a pixel $x_{i,j}$ of its receptive field on its value?

Criterion

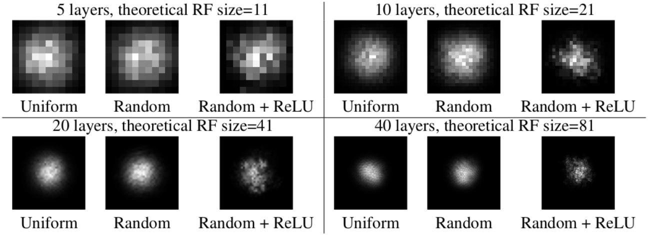
For a given output neuron y, what criterion seems sound to estimate the impact of a pixel $x_{i,j}$ of its receptive field on its value?

$$\frac{\partial y}{\partial x_{i,j}}$$

Effective receptive field

Shape of the effective receptive field

- Under some simplification hypotheses, the authors of [Luo et al., 2017] have shown that the effective receptive field has a gaussian shape.
- They have experimentally checked that this is (approximately) the case
- Moreover, as the network gets deeper, the size of the effective receptive field gets proportionally smaller



Conclusion on the receptive field

When choosing or designing a NN architecture:

- Establish its minimal required size with respect to the application
- Make it larger than that...

Contents

Equivariance

Definition

A function $f : E \longrightarrow F$ is equivariant with respect to the functions $t_E : E \longrightarrow E$ and $t_F : F \longrightarrow F$ iff $\forall x \in E$:

$$f(t_E(x)) = t_F(f(x))$$

Translation equivariance

- When $E = F$ and t_E and t_F are the same, any, translation, then we have *translation equivariance*.
- Translation equivariance is an often sought property for image processing operators.
- Note that we often abusively say *invariant* to translation instead of *equivariant* to translation.
- Given that in all practical cases images are defined on a bounded set, this property is only true "far enough" from the borders

Quizz

Which operator is NOT translation equivariant?

1. Convolution (stride$=1$)
2. Max-pooling (stride >1)
3. Transposed convolution (stride≥ 1)
4. Upsampling (stride>1)

Translation equivariant operators

Operator	**Translation equivariant**
Convolution (stride$= 1$)	
Max-pooling (stride > 1)	
Transposed convolution (stride≥ 1)	
Upsampling (stride> 1)	

Translation equivariant operators

Operator	**Translation equivariant**
Convolution (stride$=1$)	yes
Max-pooling (stride >1)	
Transposed convolution (stride≥ 1)	
Upsampling (stride>1)	

Translation equivariant operators

Operator	**Translation equivariant**
Convolution (stride$= 1$)	yes
Max-pooling (stride > 1)	no
Transposed convolution (stride≥ 1)	
Upsampling (stride> 1)	

Translation equivariant operators

Operator	**Translation equivariant**
Convolution (stride$=1$)	yes
Max-pooling (stride >1)	no
Transposed convolution (stride≥ 1)	yes
Upsampling (stride>1)	

Translation equivariant operators

Operator	**Translation equivariant**
Convolution (stride$=1$)	yes
Max-pooling (stride >1)	no
Transposed convolution (stride≥ 1)	yes
Upsampling (stride>1)	yes

Translation equivariance: comments

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

Translation equivariance: comments

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

Translation equivariance: comments

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

Translation equivariance: comments

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

Translation equivariance: comments

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

Illustration

When aiming to segment the papilla or the macula, translation equivariance is not necessarily welcome.

Credits: OPHDIAT database

Contents

Image size flexibility

- A NN containing fully-connected layers can only process images of a given size.

Image size flexibility

- A NN containing fully-connected layers can only process images of a given size.
- A fully convolutional NN can be applied to images of any size, as long as its dimensions are compatible with the subsampling steps of the network.

Image size flexibility

- A NN containing fully-connected layers can only process images of a given size.
- A fully convolutional 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 fully convolutional 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.

Contents

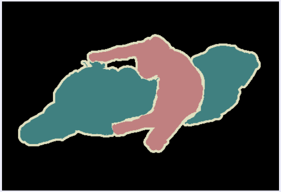
The specific case of image segmentation

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

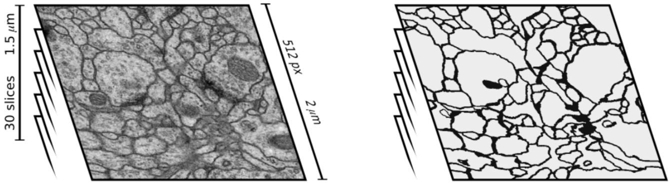
Some vocabulary on segmentation

- **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.

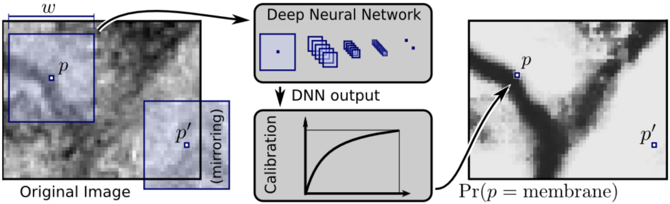
Contents

Neuron membrane segmentation challenge (ISBI 2012)

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



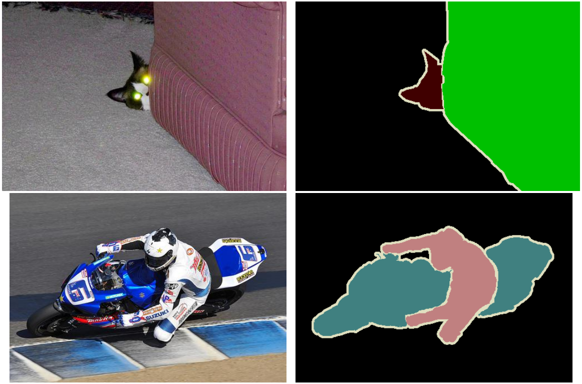
Neuron membrane segmentation challenge winner [Cireșan et al., 2012]



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]

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
- But you can reduce the number of filters or the number of layers

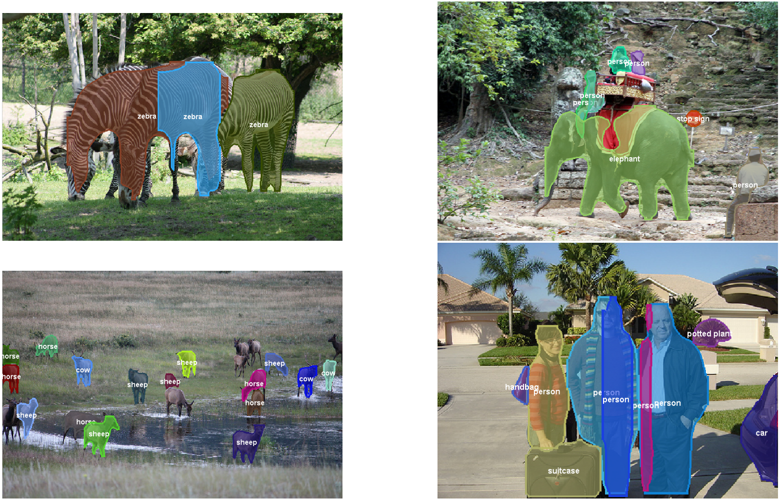
Contents

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

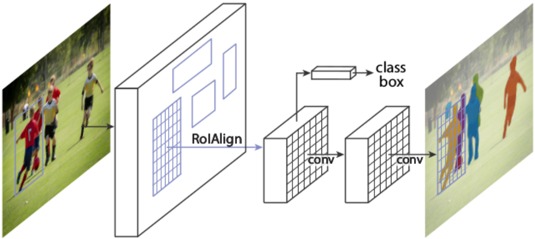
- 2 million objects, from 80 categories, in $300\,000$ images

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

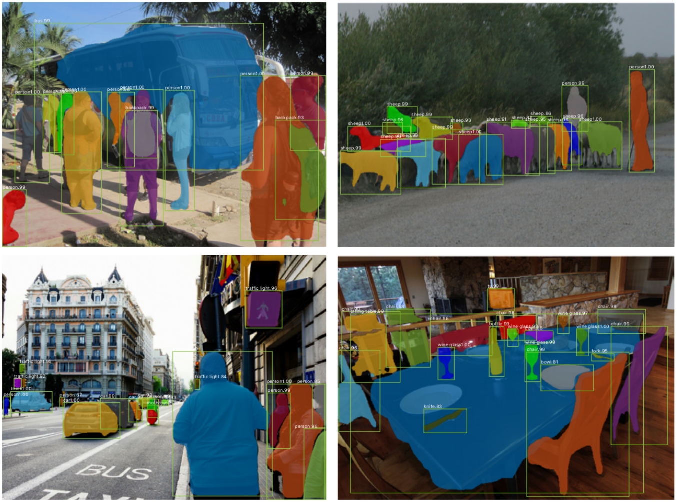
COCO instance segmentation challenge: examples of 2016 winner results



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

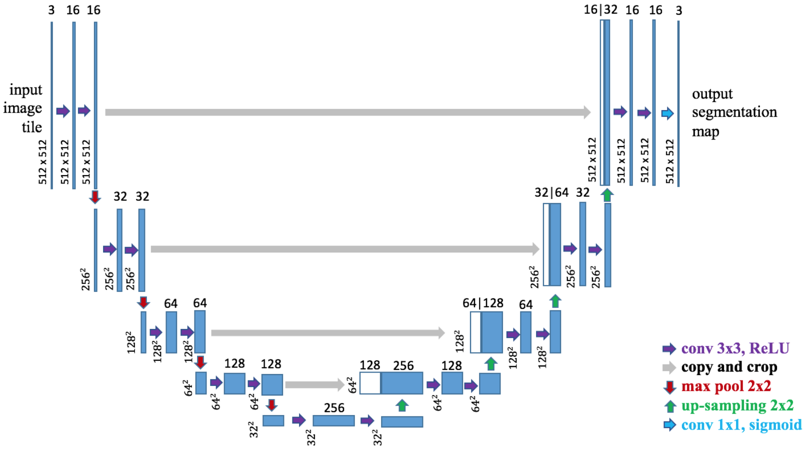


Mask R-CNN on the COCO database

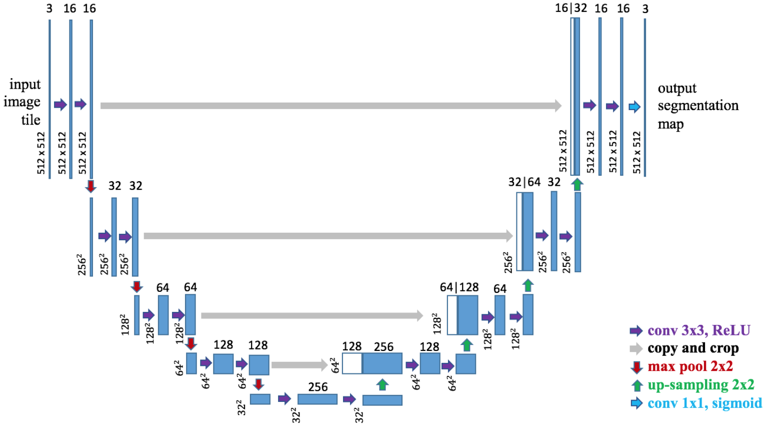


Contents

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



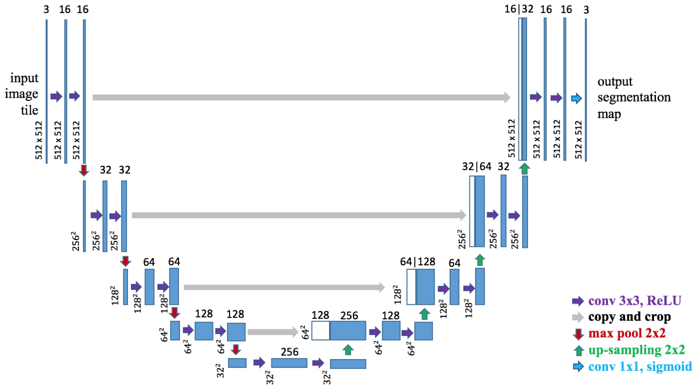
U-Net architecture [Ronneberger et al., 2015]



Quizz

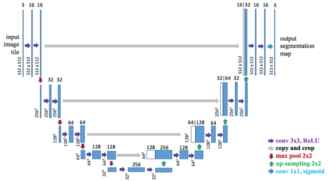
- Size and number of channels of input images?
- Segmentation into how many regions?

U-Net main ideas



- Encoding branch inspired by classification nets
- Decoding branch is symmetrical
- Skip connections

U-Net details



- 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:

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

U-Net improvements

- Convolutions with stride 2 instead of max-pooling in the encoder
- Transposed convolutions instead of simple up-sampling in the decoder
- Batch normalization
- Using an already optimized classification network as backbone for the encoder

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 equivariance 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 equivariance 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 equivariance 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

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

A loss function inherited from image classification

- Cross-entropy: $-\sum_i y_i \log(\hat{y}_i)$

Measures used in image processing

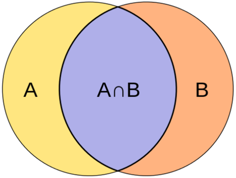
Let A and B be two sets, not simultaneously empty.

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 $[0, 1]$

Jaccard index

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

But $\mathbf{y}$ and $\hat{\mathbf{y}}$ are in $[0, 1]^n \dots$

Generalization to $[0,1]$

Jaccard index

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

But $\mathbf{y}$ and $\hat{\mathbf{y}}$ are in $[0,1]^n$ ...

Jaccard similarity

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

($\mathbf{y}$ and $\hat{\mathbf{y}}$ are not simultaneously equal to 0)

Corresponding loss function

Jaccard loss

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

Constant ϵ, which is typically "small", keeps the denominator "far enough" from zero.

Corresponding loss function - variant

Jaccard loss

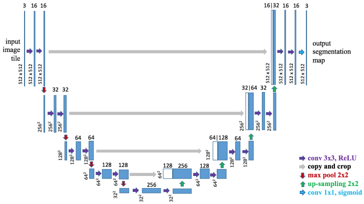
$$j_2(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{\sum_i y_i \hat{y}_i}{\sum_i y_i^2 + \sum_i \hat{y}_i^2 - \sum_i y_i \hat{y}_i + \epsilon}$$

This version works slightly better than the first [Duque-Arias et al., 2021].

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 is an interesting research path.

Applying fully-convolutional networks



Case study

Suppose that we have satisfactorily optimized this U-Net model, using images of size 512×512 during training. Now I want to apply this model to a new image, of size 1000×1000. How should I proceed?

1. Resize the image?
2. Cut it into 512×512 crops, predict and stitch the results together?
3. Pad the image.

Contents

Segment anything

https://ai.facebook.com/blog/segment-anything-foundation-model-image-segmentation/

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.

[Cireşan et al., 2012] Cireşan, 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.

[Duque-Arias et al., 2021] Duque-Arias, D., Velasco-Forero, S., Deschaud, J.-E., Goulette, F., Serna, A., Decencière, E., and Marcotegui, B. (2021). On power Jaccard losses for semantic segmentation. In *VISAPP 2021 : 16th International Conference on Computer Vision Theory and Applications*, Vienne (on line), Austria.

[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.

[Feng Ning et al., 2005] Feng Ning, Delhomme, D., LeCun, Y., Piano, F., Bottou, L., and Barbano, P. E. (2005). Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing*, 14(9):1360–1371. Conference Name: IEEE Transactions on Image Processing.

References II

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

[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.

[Jain et al., 2007] Jain, V., Murray, J. F., Roth, F., Turaga, S., Zhigulin, V., Briggman, K. L., Helmstaedter, M. N., Denk, W., and Seung, H. S. (2007). Supervised Learning of Image Restoration with Convolutional Networks. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. ISSN: 2380-7504.

[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.

[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

[Luo et al., 2017] Luo, W., Li, Y., Urtasun, R., and Zemel, R. (2017). Understanding the Effective Receptive Field in Deep Convolutional Neural Networks. *arXiv:1701.04128 [cs]*. arXiv: 1701.04128.

[Ounkomol et al., 2018] Ounkomol, C., Seshamani, S., Maleckar, M. M., Collman, F., and Johnson, G. R. (2018). Label-free prediction of three-dimensional fluorescence images from transmitted light microscopy. *Nature methods*, 15(11):917–920.

[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.

[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.