Deep neural networks for segmentation.

Application to Histopathology

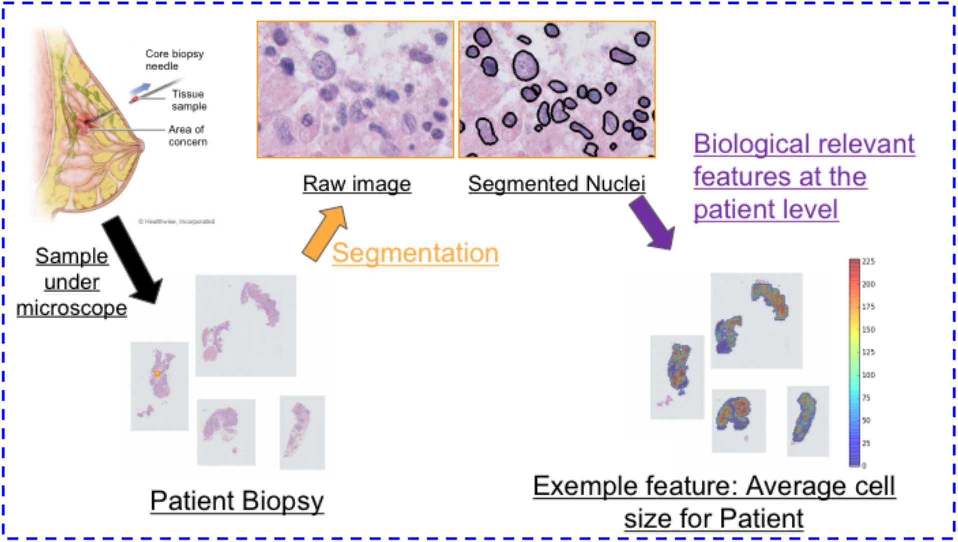
Peter Naylor, Marick Laé, Fabien Reyal and Thomas Walter

Friday, 29th June 2018

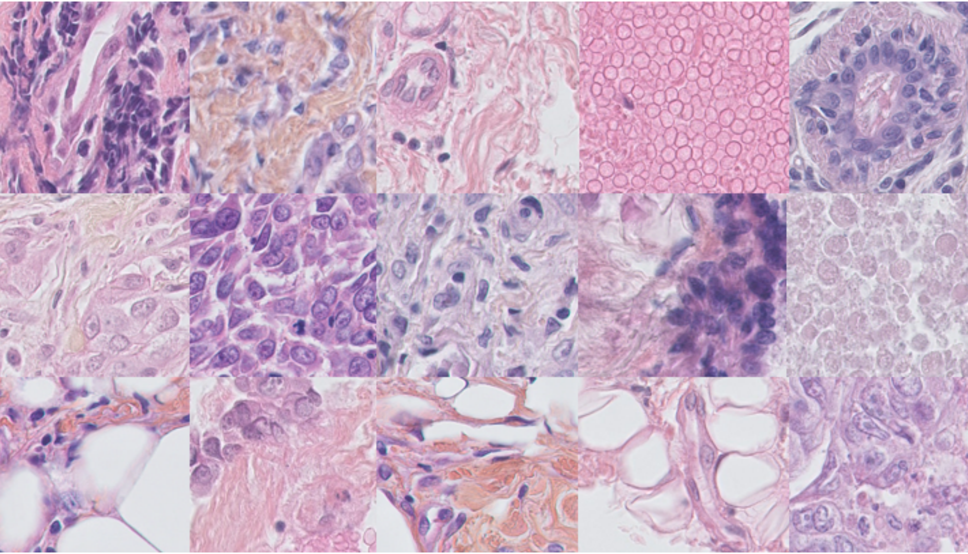
Overview

Summary

Nuclei segmentation: why should you care?

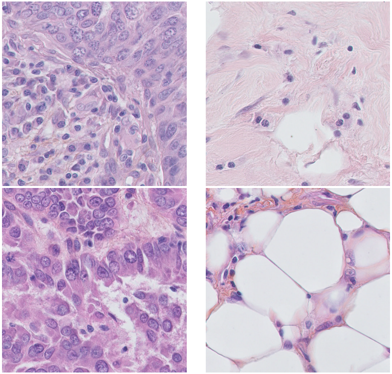


Nuclei segmentation: challenging task



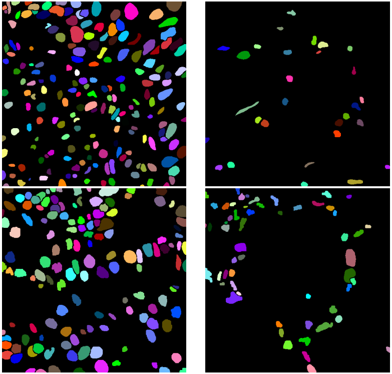
Supervised learning

Manual annotation - *DS1*



Raw data

- TNBC patients
- 50 annotated images
- 11 patients
- 4022 annotated nuclei

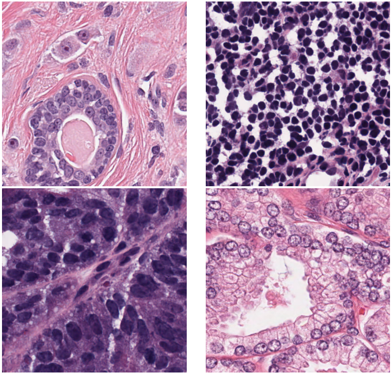


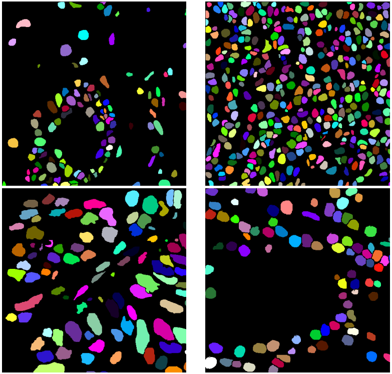
Ground truth

- $y \in \{0, 1\}^{height \times width}$
- We do a pixel wise prediction
- `PeterJackNaylor.github.io`

Supervised learning

DS2 - [Kumar et Al 2017]





Raw data

- Patients samples provided by TCGA
- 30 annotated images from different patients
- From 6 different organs
- 21 623 annotated nuclei

Ground truth

- We split in train (12), validation (4), test (14)
- Comparaison on F1 and AJI
- `nucleisegmentationbenchmark.weebly.com/`

Metrics

F1 pixel metric:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where Recall $= \frac{TP}{TP+FN}$ and Precision $= \frac{TP}{TP+FP}$.

AJI object metric:

$G = \bigcup_{i=1\ldots L} G_i$, the ground truth where G_i is the ground truth set pixels belonging to the the i-th connected component.
$S = \bigcup_{k=1\ldots M} S_k$, the detection results where S_k is k-th connected component.

$$AJI = \frac{\sum_{i=1}^{L} |G_i \cap S_k^*(i)|}{\sum_{i=1}^{L} |G_i \cup S_k^*(i)| + \sum_{l \in U} |S_l|} \tag{1}$$

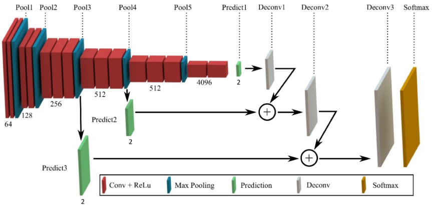
where $S_k^*(i) = \arg\max_{S_\nu} \frac{|G_i \cap S_\nu|}{|G_i \cup S_\nu|}$ (with $S_k^*(i) \cap S_l^*(j) = \emptyset$ for $i \neq j$) and U is the set of indices of all unassigned detections.

Summary

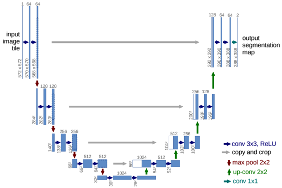
Methods for segmentation

Pixel wise classification. Many possible designs:

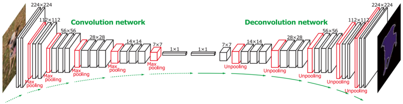
- Thresholding and mathematical morphology
- Series of convolution layers, PangNet [Pang et Al 2010]
- Sliding window (with any classifier)
- FCN [Long et Al 2014]
- DeconvNet [Noh et Al 2015]
- U-Net [Ronneberger et Al 2015]
- R-CNN [He et Al 2018]



FCN



UNet

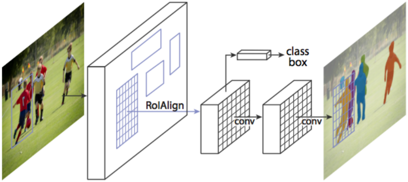


DeconvNet

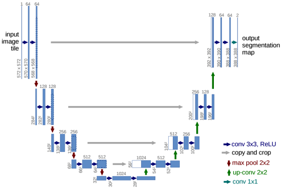
Methods for segmentation

Pixel wise classification. Many possible designs:

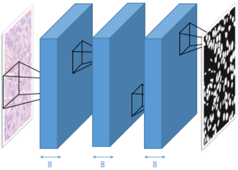
- Thresholding and mathematical morphology
- Series of convolution layers, PangNet [Pang et Al 2010]
- Sliding window (with any classifier)
- FCN [Long et Al 2014]
- DeconvNet [Noh et Al 2015]
- U-Net [Ronneberger et Al 2015]
- R-CNN [He et Al 2018]



R-CNN



UNet



PangNet

Convolutional neural networks

Convolutional layer

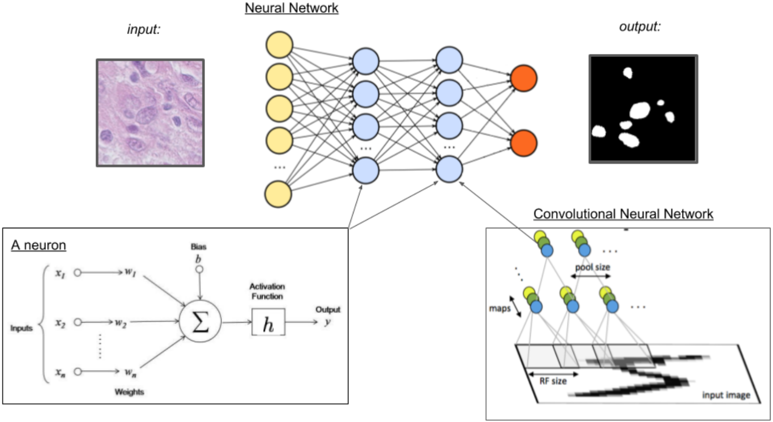


Figure: Convolutional Neural Networks

Convolutional neural networks

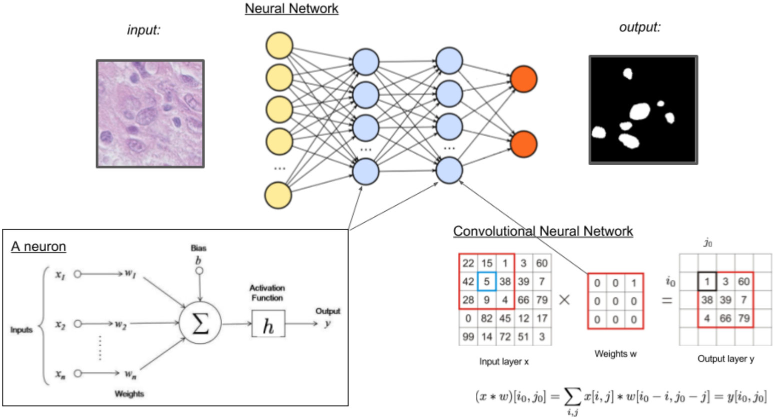
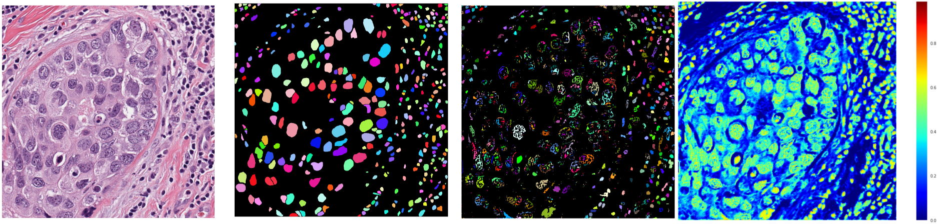


Figure: Convolutional Neural Networks

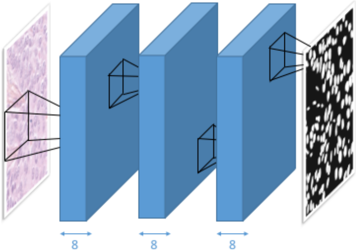
PangNet



Raw Image *Ground Truth* *Prediction* *Probability*

- 8 feature maps
- 3 hidden layers
- Y is the output/probability maps
- We learn (3X3)x3x8 + 3 parameters in the first layer, (3X3)x8x8 + 8 parameters in the second layer ..
-

	PangNet
F1 (Pixel)	0.5614
AJI (Object)	0.2486



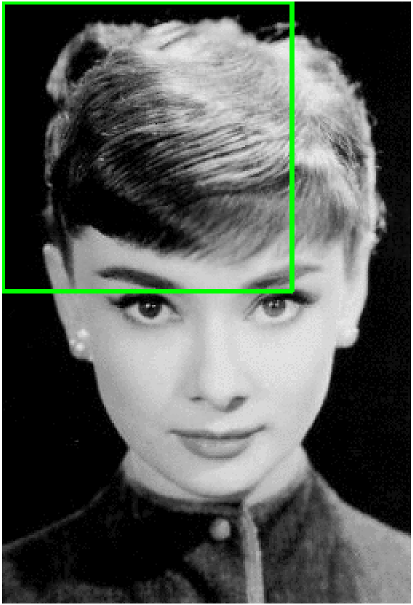
Architecture [Pang et Al 2010]

Sliding window

- Any classifier can be used
- Per pixel prediction: computationnaly intensive
- First use of Neural Networks for segmentation
- Still used today as the networks can be much smaller

Layer	Filter size	Activation	Output size	Dropout rate
Input	–	–	$51 \times 51 \times 3$	–
Conv 1	4×4	ReLU	$48 \times 48 \times 25$	0.1
Pool 1	2×2	Max	$24 \times 24 \times 25$	–
Conv 2	5×5	ReLU	$20 \times 20 \times 50$	0.2
Pool 2	2×2	Max	$10 \times 10 \times 50$	–
Conv 3	6×6	ReLU	$5 \times 5 \times 80$	0.25
Pool 3	2×2	Max	$3 \times 3 \times 80$	–
FC 1	–	ReLU	1024	0.5
FC 2	–	ReLU	1024	0.5
Output	–	SoftMax	3	–

Figure: Example of a sliding window network [Kumar et Al 2017]



Summary

Fully Convolutional Networks [Long et Al 2014]

Some remarks:

- End to end training of the neural network
- Fully-connected layers are "only convolution layers with 1x1 convolution kernels"
  You can feed any size image and the same sized output
- Success of pre-training, for the encoder part, the "What"
- Decoder: repositioning the content from the encoder, the "Where"

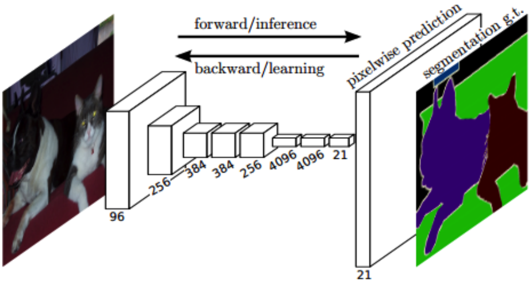


Figure: Convolutional Neural Networks

Fully Convolutional Networks [Long et Al 2014]

Some remarks:

- End to end training of the neural network
- Fully-connected layers are "only convolution layers with 1x1 convolution kernels"
  You can feed any size image and the same sized output
- Success of pre-training, for the encoder part, the "What"
- Decoder: repositioning the content from the encoder, the "Where"

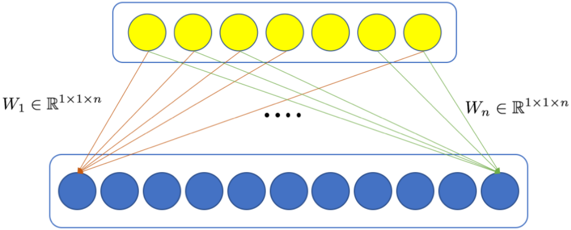


Figure: Convolutional Neural Networks

Fully Convolutional Networks [Long et Al 2014]

Some remarks:

- End to end training of the neural network
- Fully-connected layers are "only convolution layers with 1x1 convolution kernels"
  You can feed any size image and the same sized output
- Success of pre-training, for the encoder part, the "What"
- Decoder: repositioning the content from the encoder, the "Where"

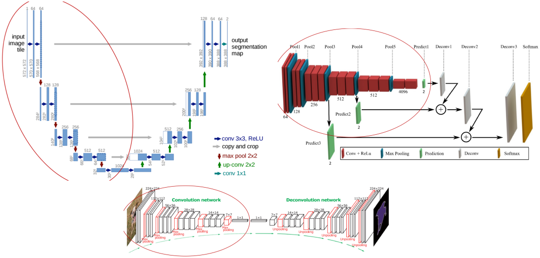


Figure: Convolutional Neural Networks

Fully Convolutional Networks [Long et Al 2014]

Some remarks:

- End to end training of the neural network
- Fully-connected layers are "only convolution layers with 1x1 convolution kernels"
  You can feed any size image and the same sized output
- Success of pre-training, for the encoder part, the "What"
- Decoder: repositioning the content from the encoder, the "Where"

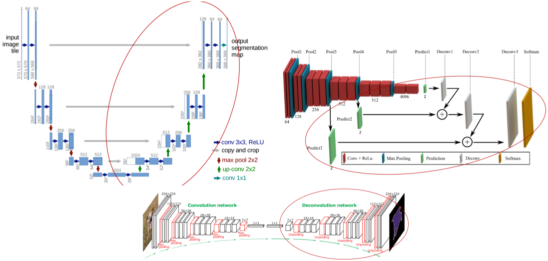


Figure: Convolutional Neural Networks

U-net

- Won many challenges: ISBI EM stacks 2012, ISBI cell tracking challenge 2015
- Can be trained from scratch
- Skip layers
- Encoder, the "What" , Decoder, the "Where"
- Outputs a probability map

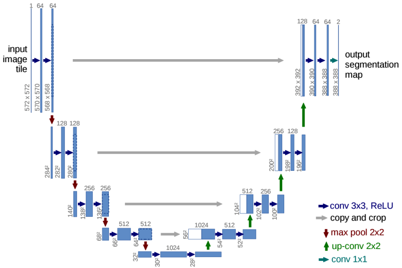


Figure: U-net [Ronneberger et Al 2015]

U-net

- Won many challenges: ISBI EM stacks 2012, ISBI cell tracking challenge 2015
- Can be trained from scratch
- Skip layers
- Encoder, the "What" , Decoder, the "Where"
- Outputs a probability map

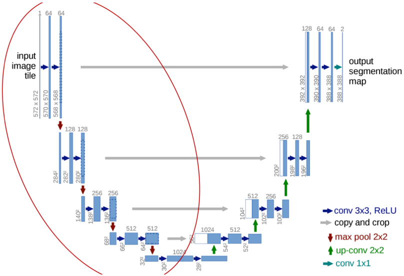


Figure: U-net [Ronneberger et Al 2015]

U-net

- Won many challenges: ISBI EM stacks 2012, ISBI cell tracking challenge 2015
- Can be trained from scratch
- Skip layers
- Encoder, the "What" , Decoder, the "Where"
- Outputs a probability map

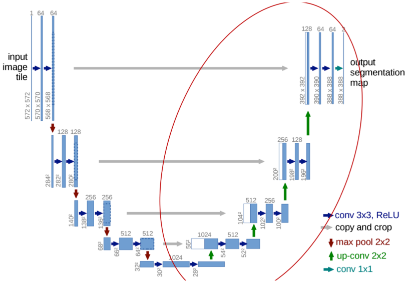
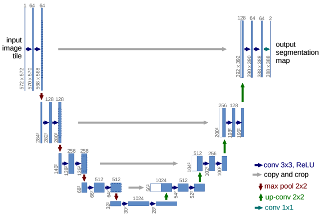


Figure: U-net [Ronneberger et Al 2015]

U-net

- Won many challenges: ISBI EM stacks 2012, ISBI cell tracking challenge 2015
- Can be trained from scratch
- Skip layers
- Encoder, the "What" , Decoder, the "Where"
- Outputs a probability map



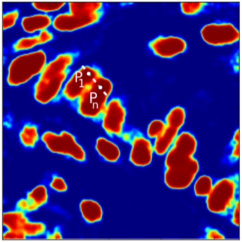


Figure: U-net [Ronneberger et Al 2015]

Fully Convolutional Networks to nuclei segmentation

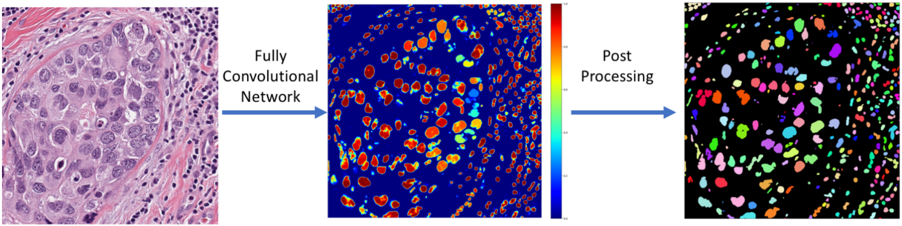


Figure: Typical pipeline

Fully Convolutional Networks to nuclei segmentation

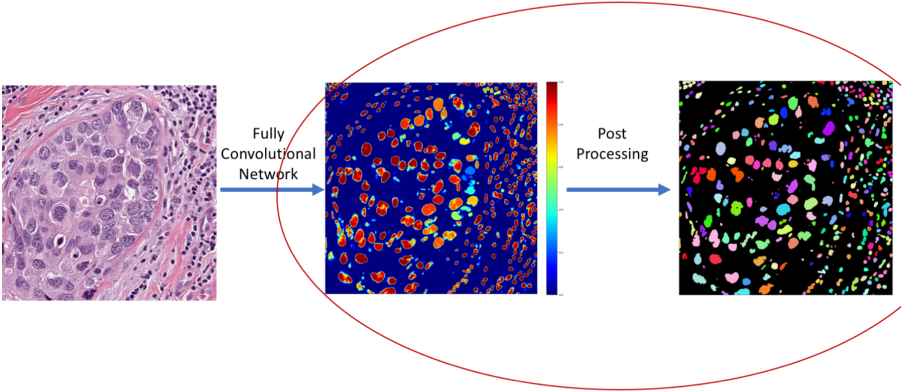


Figure: Post-processing

Fully Convolutional Networks to nuclei segmentation

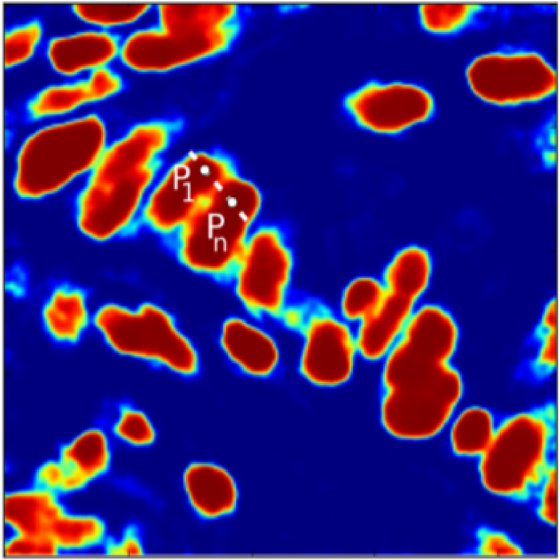


Figure: Typical probability map of a trained fully convolutional network

Fully Convolutional Networks to nuclei segmentation

- The posterior probability at the nucleus border is systematically lower than in the putative center of the nucleus
- Let $\mathcal{P}$ be the all possible probability pathways from two candidates P_1 and P_N (two maximum a posteriori). We split if:

$$\min_{\mathcal{P}} C(\mathcal{P}) = \min_{\mathcal{P}} \{ \max_{i=2...N} p_1 - p_i \} > \lambda$$

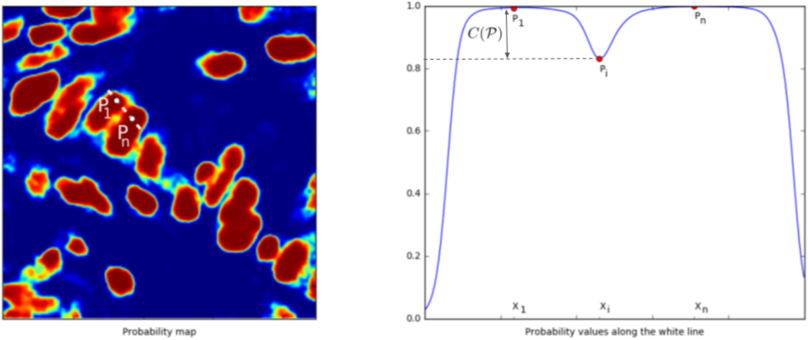
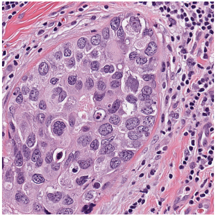
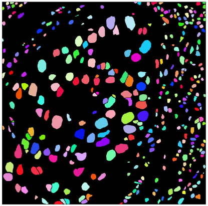


Figure: Values along one arbitrary path between two candidates.

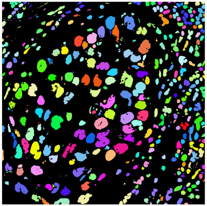
U-net



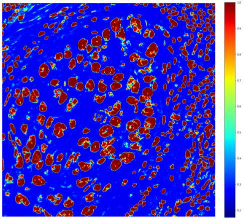
Raw Image



Ground Truth



Prediction



Probability

	PangNet	U-net	U-net + PP
F1 (Pixel)	0.5614	0.7793	0.7793
AJI (Object)	0.2486	0.3833	0.5182

Table: Summary results

Summary

Seperating touching nuclei

Litterature

- Some add a 3rd class border pixels [Kumar et Al 2015]
- Weighting inter-object pixels [Ronneberger et Al 2015]
- Instance segmentation: RCNN [He et Al 2018]
- The post-processing seperates touching nuclei with respect to the probability distribution, why not predict something better for the post-processing?

Turning a classification into a regression problem

Changing the task

- Pixelwise binary classification
- $y \in \{0,1\}^{height \times width}$

$$loss(y, \widehat{y}) = \sum_{i,j} \sum_{k} t_{i,j,k} \log(p_{i,j,k})$$

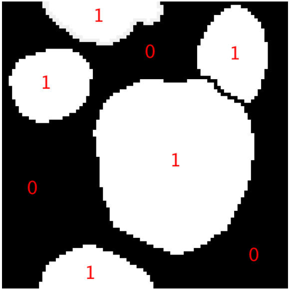


Figure: y

- Pixelwise regression
- $y^D \in \mathbb{R}^{height \times width}$
  $y^D = DistanceMap(y)$

$$loss(y^D, \widehat{y^D}) = \frac{1}{np} \sum_{i,j} (y^D_{i,j} - \widehat{y^D_{i,j}})^2$$

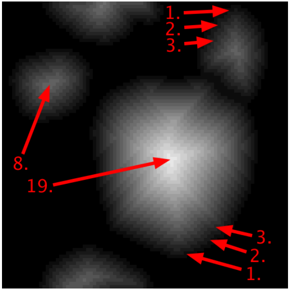


Figure: y^D

Turning a classification into a regression problem

Changing the task

- Pixelwise binary classification
- $y \in \{0,1\}^{height \times width}$

$$loss(y, \widehat{y}) = \sum_{i,j} \sum_{k} t_{i,j,k} \log(p_{i,j,k})$$

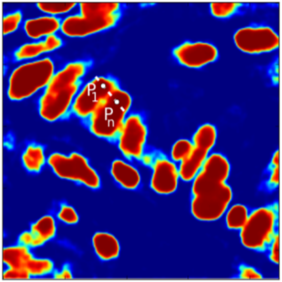


Figure: p

- Pixelwise regression
- $y^D \in \mathbb{R}^{height \times width}$
  $y^D = DistanceMap(y)$

$$loss(y^D, \widehat{y^D}) = \frac{1}{np} \sum_{i,j} (y^D_{i,j} - \widehat{y^D_{i,j}})^2$$

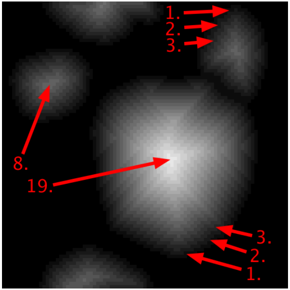
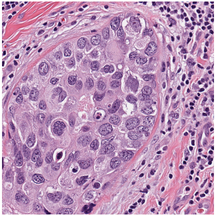
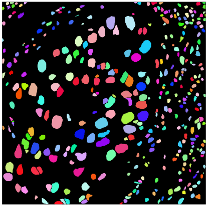
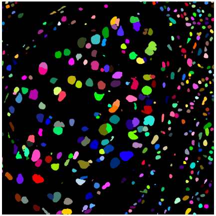


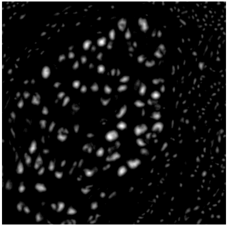
Figure: y^D

Distance Regression


Raw Image


Ground Truth


Prediction


Distance map

	PangNet	U-net	U-net + PP	DIST
F1 (Pixel)	0.5614	0.7793	0.7793	0.7863
AJI (Object)	0.2486	0.3833	0.5182	0.5598

Table: Summary results

Summary

R-CNN

Chronologically:

1. R-CNN [Girshick et Al 2014]
2. Fast R-CNN [Girshick et Al 2015]
3. Faster R-CNN: - Speeding Up Region Proposal [Ren et Al 2015]
4. Mask R-CNN [He et Al 2018]

What a R-CNN is:

1. Generate regions of interest, bounding boxes.
2. Classify the bouding box.
3. Correct the bounding box via a linear regression once the object is classified
4. (optionnaly) Generate a mask in parallel to the classification and to the correction of the bounding box

R-CNN

1. R-CNN [Girshick et Al 2014]
2. Fast R-CNN [Girshick et Al 2015]
3. Faster R-CNN: - Speeding Up Region Proposal [Ren et Al 2015]
4. Mask R-CNN [He et Al 2018]

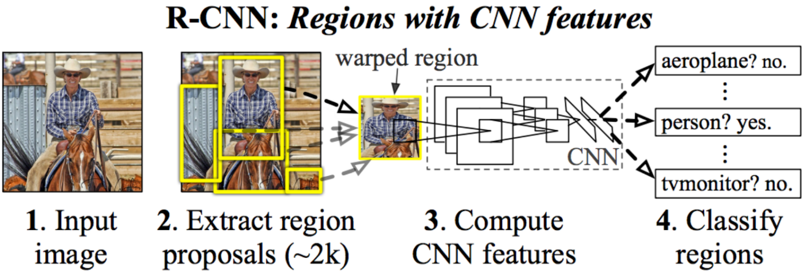


Figure: [Girshick et Al 2014]

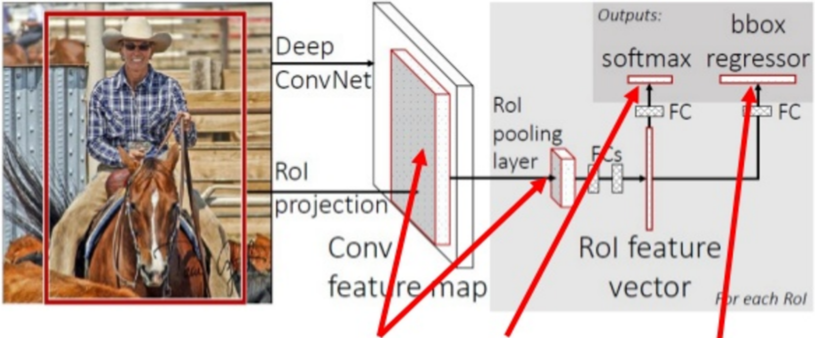
R-CNN

1. R-CNN [Girshick et Al 2014]
2. Fast R-CNN [Girshick et Al 2015]
3. Faster R-CNN: - Speeding Up Region Proposal [Ren et Al 2015]
4. Mask R-CNN [He et Al 2018]

- RoI (Region of Interest) pooling: one run per image, and not per proposed RoI
- Joint training framework

R-CNN

1. R-CNN [Girshick et Al 2014]
2. Fast R-CNN [Girshick et Al 2015]
3. Faster R-CNN: - Speeding Up Region Proposal [Ren et Al 2015]
4. Mask R-CNN [He et Al 2018]



Joint the feature extractor, classifier, regressor together in a unified framework

R-CNN

1. R-CNN [Girshick et Al 2014]
2. Fast R-CNN [Girshick et Al 2015]
3. Faster R-CNN: - Speeding Up Region Proposal [Ren et Al 2015]
4. Mask R-CNN [He et Al 2018]

- Removed Selective Search algorithm for the RoI proposals
- A single CNN is used end to end

R-CNN

1. R-CNN [Girshick et Al 2014]
2. Fast R-CNN [Girshick et Al 2015]
3. Faster R-CNN: - Speeding Up Region Proposal [Ren et Al 2015]
4. Mask R-CNN [He et Al 2018]

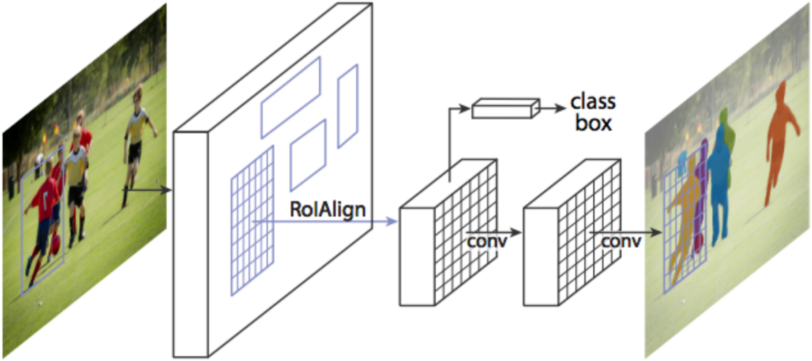
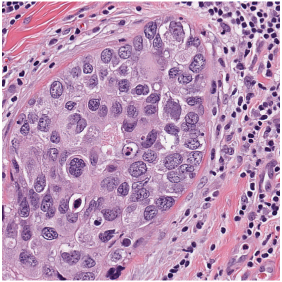
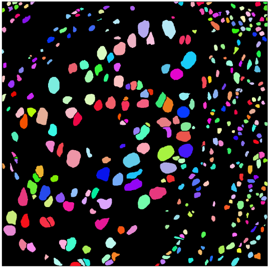


Figure: [He et Al 2018]

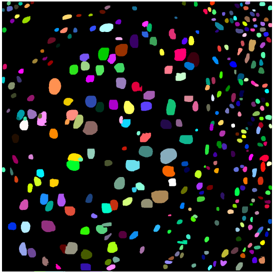
R-CNN



Raw Image



Ground Truth



Prediction

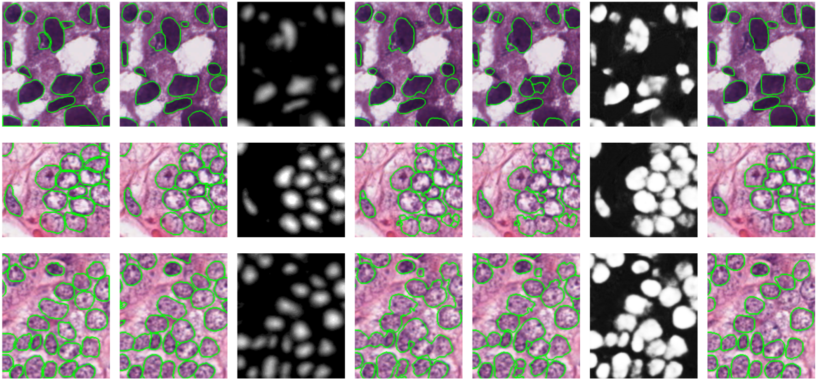
	PangNet	U-net	U-net + PP	DIST	RCNN
F1 (Pixel)	0.5614	0.7793	0.7793	0.7863	0.7470
AJI (Object)	0.2486	0.3833	0.5182	0.5598	0.5002

Table: Summary results

Result Summary

	PangNet	U-net	U-net + PP	DIST	RCNN
F1 (Pixel)	0.5614	0.7793	0.7793	0.7863	0.7470
AJI (Object)	0.2486	0.3833	0.5182	0.5598	0.5002

Table: Summary results



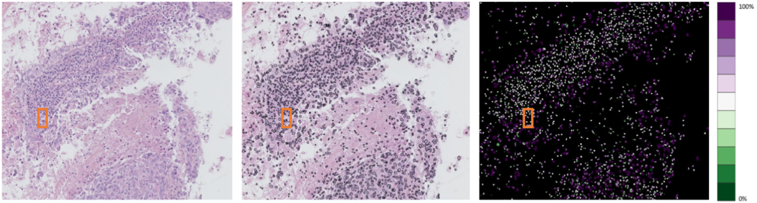
(a) Ground Truth (b) DIST (c) Distance regression output (d) U-Net (e) U-Net + PP (f) U-Net probability map (g) RCNN

Summary

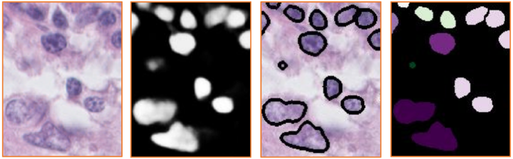
Conclusion

- The state of the art in segmentation are fully convolutional neural network: namely U-net and Mask R-CNN.
- Pooling helps the model have a better notion of the objects.
- Best results in computer vision for a given task are a mix between deep neural networks and traditionnal methods.

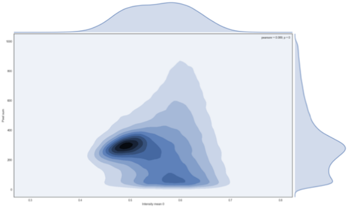
Perspectives: Profiling patients by their cell population



Raw image *Segmented* *Pixel Size*

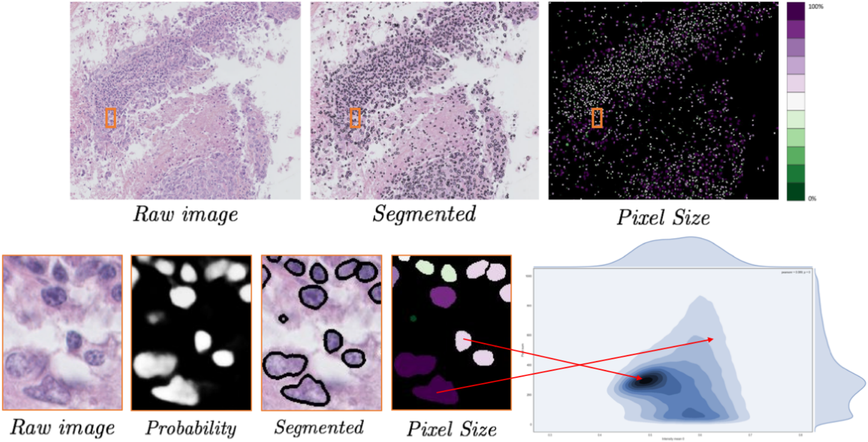


Raw image *Probability* *Segmented* *Pixel Size*



Histogram plot of size and intensity for the cell population

Perspectives: Profiling patients by their cell population



Histogram plot of size and intensity for the cell population

References

- [Pang et Al 2010] "Cell nucleus segmentation in color histopathological imagery using convolutional networks," *in Pattern Recognition (CCPR), 2010 Chinese Conference on.* IEEE, 2010, pp. 1–5.
- [Long et Al 2014] "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015*, pp. 3431–3440.
- [Ronneberger et Al 2015] "U-Net: Convolutional Networks for Biomedical Image Segmentation", *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.
- [Kumar et Al 2017] "A Dataset and a Technique for Generalized Nuclear Segmentation for Computational Pathology". In *IEEE Transactions on Medical Imaging*, 2017.
- [Naylor et Al 2017] "Nuclei Segmentation in Histopathology Images using Deep Neural Networks". In *International Symposium on Biomedical Imaging (ISBI'17), Melbourne, Australia*, April 18-21, 2017.

References

- [Girshick et Al 2014] "Rich feature hierarchies for accurate object detection and semantic segmentation", *Proceedings of the IEEE conference on computer vision and pattern recognition. 2014* IEEE, 2014
- [Girshick et Al 2015] "Fast R-CNN", in *arXiv preprint arXiv:1504.08083 (2015).*
- [Ren et Al 2015] " Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in neural information processing systems.* 2015.
- [He et Al 2018] "Nuclei Segmentation in Histopathology Images using Deep Neural Networks". In *arXiv preprint arXiv: 170306870."* (2017)

Acknowledgements

Thank you and thanks to:

- CBIO

PI: C. Azencott, V. Stoven, T. Walter and J-P. Vert.

PhD and Post-Doc: J. Abecassis, Y. Jiao, B. Khalfaoui, M. Le Morvan, B. Playe, H. Climente-Gonzalez, L. Slim and J. Boyd.

IT departement and I. Schmitt.

- U900
- Service de Pathologie, Institut Curie

In particular: M. Laé, A. Nicolas.

- Ligue Nationnal Contre le Cancer

CBIO Team

MINES ParisTech

institut Curie

Inserm
Institut national de la santé et de la recherche médicale

LA LIGUE CONTRE LE CANCER