Deep learning for medical imaging

Olivier Colliot, PhD
Research Director at CNRS
Co-Head of the ARAMIS Lab – www.aramislab.fr
PRAIRIE – Paris Artificial Intelligence Research Institute

Maria Vakalopoulou, PhD
Assistant Professor at CentraleSupelec
Mathematics and Informatics (MICS)
Office: Bouygues Building Sb.132

Master 2 - MVA

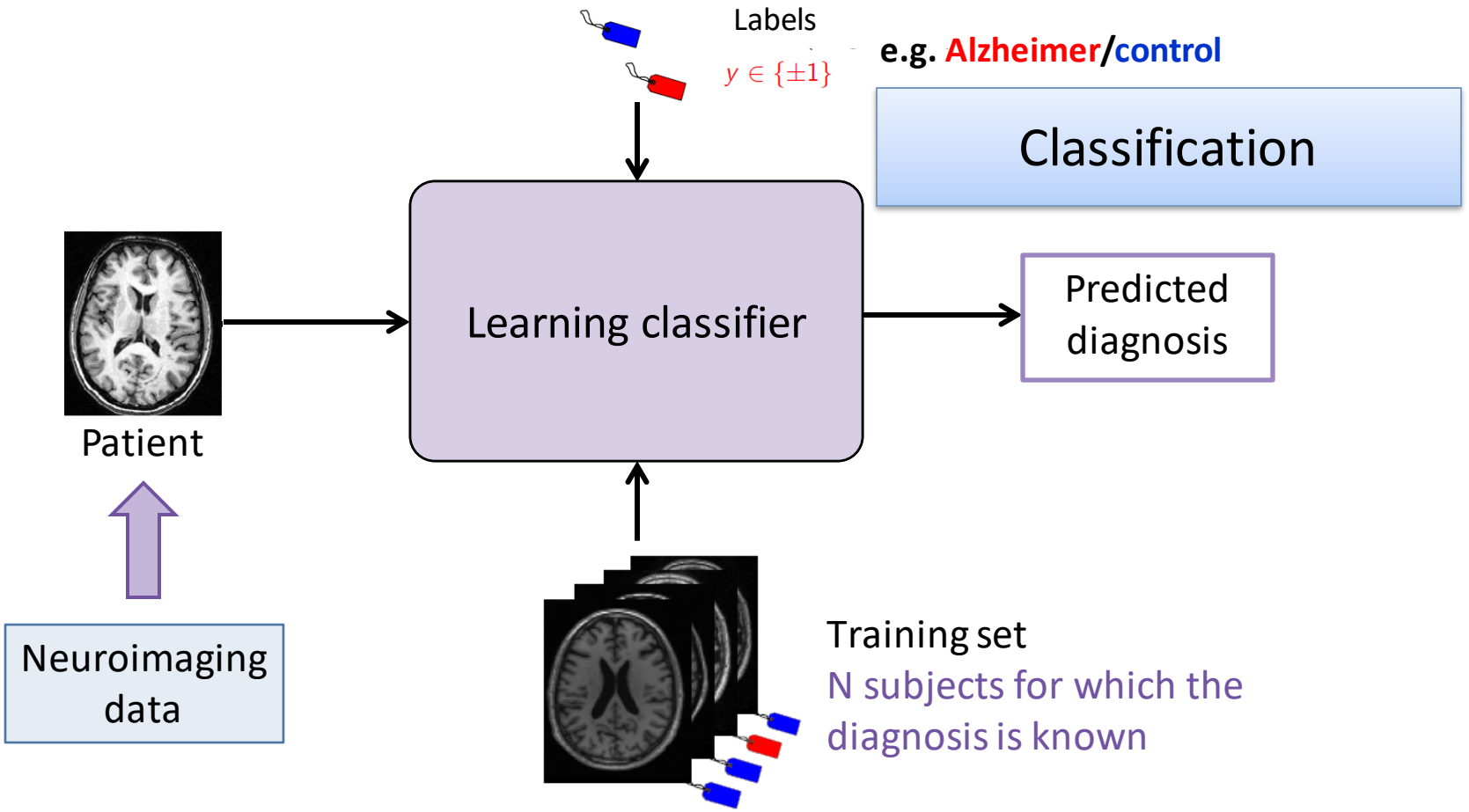
Acknowledgements

- The lecture is partially base on material by:
  - Andrej Karpathy
  - Fei-Fei Li
  - Daniel Rueckert
  - Idan Bassuk
  - Ross Girshick

Thank you!!

Previous Lectures

Classification



Classification and Regression

Input variable (multivariate): $\boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix}$

Input features

Output: y

Model: $f,\ y = f(x)$

The "artificial intelligence"

Loss: $\ell(y, x)$

Quantifies how much the prediction is far from the true output

Cost function:

$$J(f) = \frac{1}{n} \sum_{i=1}^{n} \ell \left(y^{(i)}, f(x^{(i)}) \right)$$

How far are we from the true output across all training examples ?

Learning:

$$\hat{f} = arg \min_{f \in \mathcal{F}} J(f)$$

Learning: find the model with the minimal error

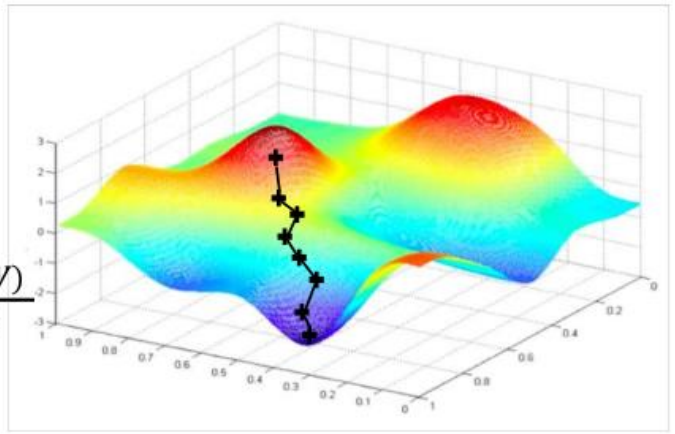
Optimization algorithm: method to find the minimum

Stochastic gradient descent

Stochastic gradient descent with several samples (mini-batch)

Algorithm

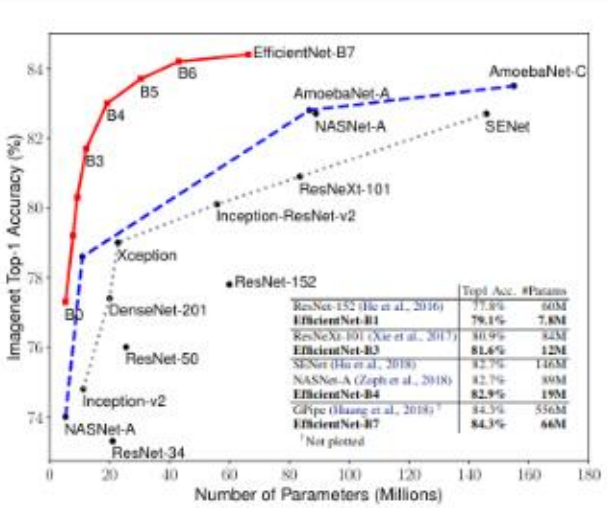
1. Initialize weights randomly $\sim \mathcal{N}(0, \sigma^2)$
2. Loop until convergence:
3. Pick batch of B data points
4. Compute gradient, $\frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}} = \frac{1}{B}\sum_{k=1}^{B} \frac{\partial J_k(\boldsymbol{W})}{\partial \boldsymbol{W}}$
5. Update weights, $\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta \frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}$
6. Return weights

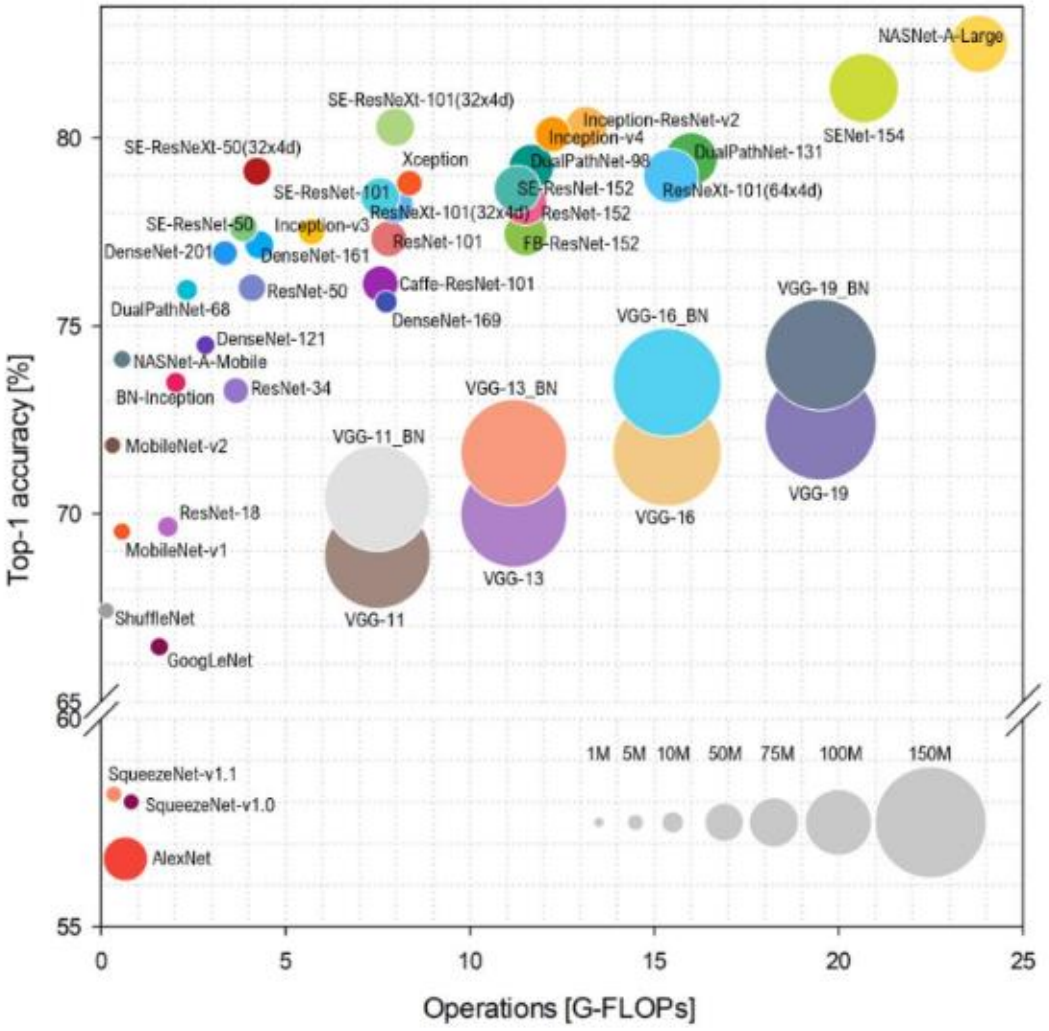


Fast to compute and a much better estimate of the true gradient!

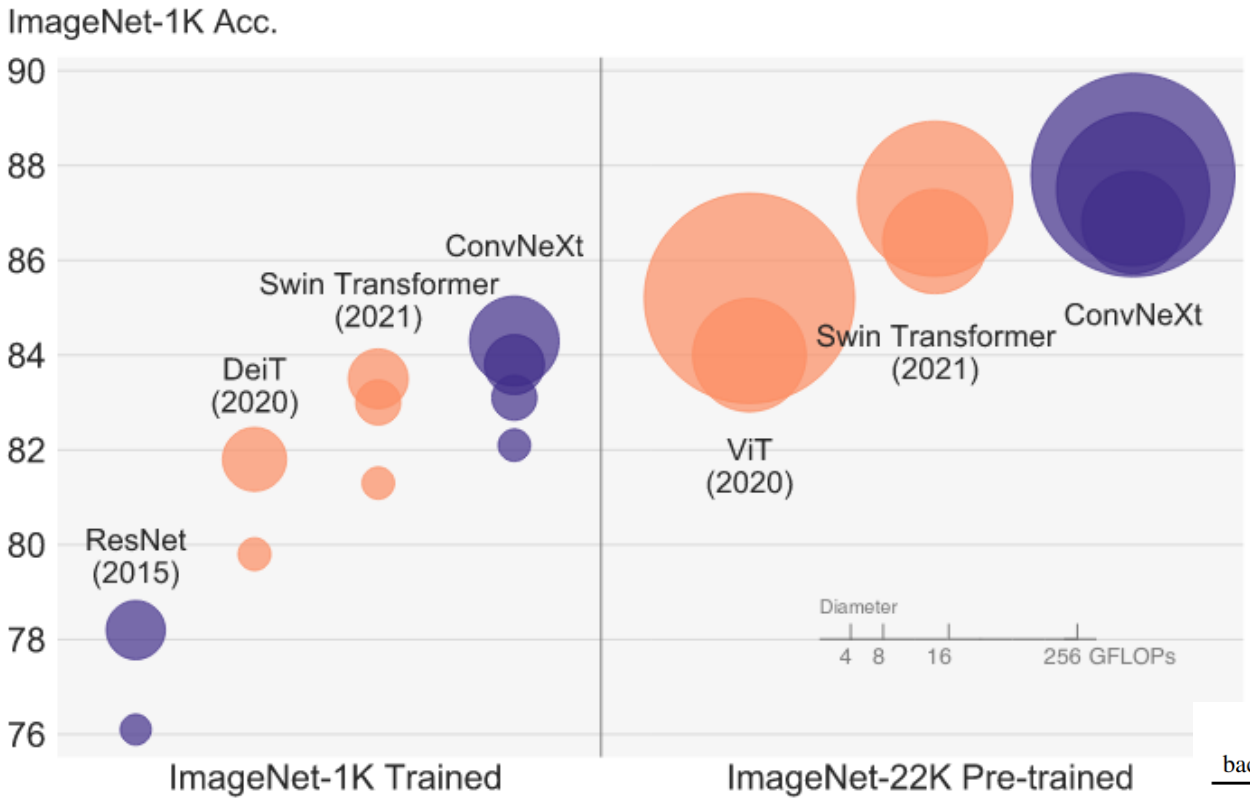
Source: http://introtodeeplearning.com/

Common architectures





Different types of architectures



backbone	input crop.	mIoU	#param.	FLOPs
ImageNet-1K pre-trained				
○ Swin-T	512^2	45.8	60M	945G
● ConvNeXt-T	512^2	**46.7**	60M	939G
○ Swin-S	512^2	49.5	81M	1038G
● ConvNeXt-S	512^2	**49.6**	82M	1027G
○ Swin-B	512^2	49.7	121M	1188G
● ConvNeXt-B	512^2	**49.9**	122M	1170G
ImageNet-22K pre-trained				
○ Swin-B‡	640^2	51.7	121M	1841G
● ConvNeXt-B‡	640^2	**53.1**	122M	1828G
○ Swin-L‡	640^2	53.5	234M	2468G
● ConvNeXt-L‡	640^2	**53.7**	235M	2458G
● ConvNeXt-XL‡	640^2	**54.0**	391M	3335G

A ConvNet for the 2020s, arXiv:2201.03545v1

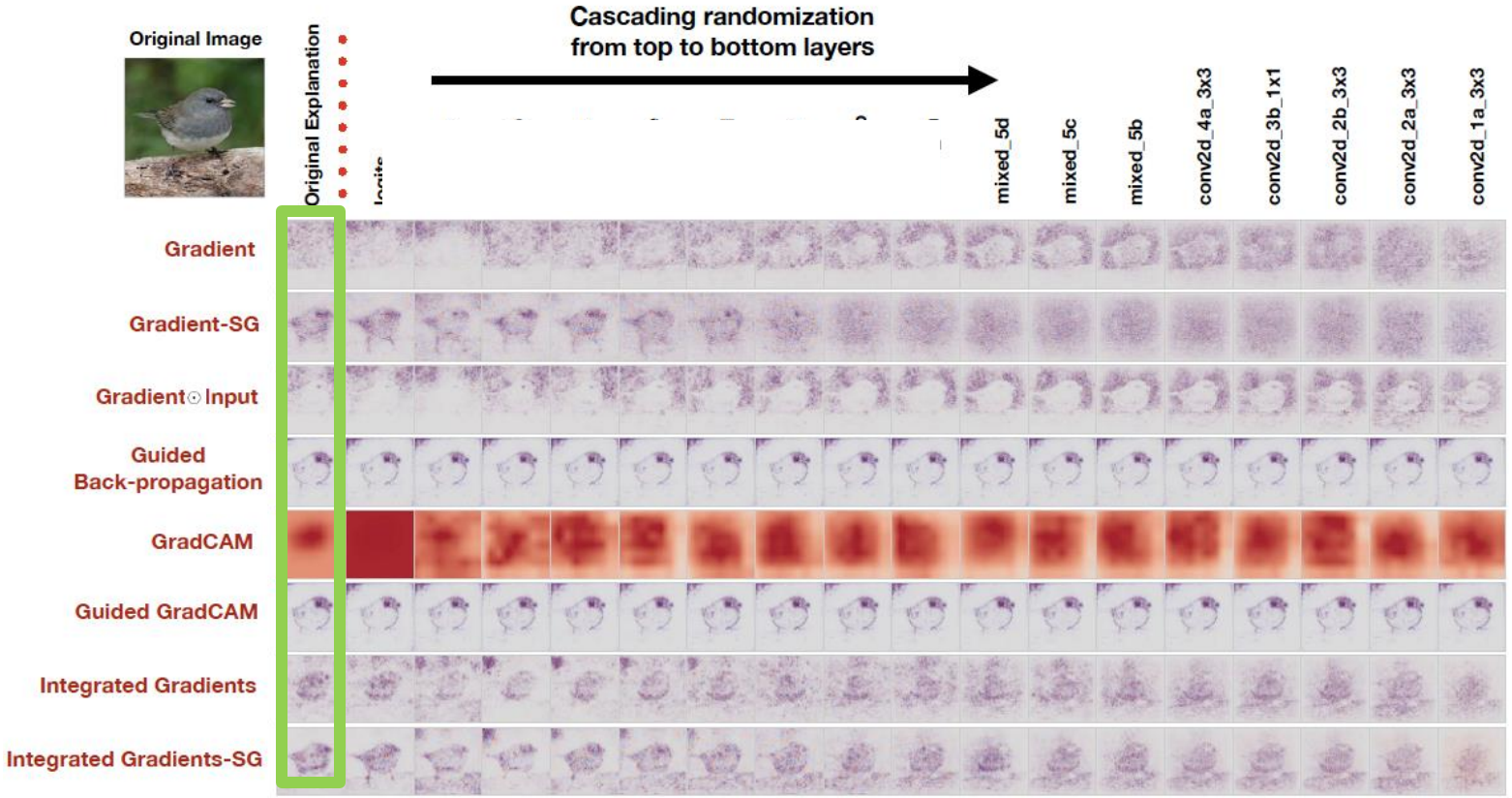
Metrics for classification

Evaluation

- **Different metrics provide complementary information**

- **Recommandations**
  - **Always look at all the individual metrics**
    - Sensitivity, Specificity, NPV, PPV
  - **Never trust a single aggregate metric** (accuracy, BA, AUC…)
  - Add multiple aggregate metrics:
    - Accuracy, BA, F1, MCC
  - **Understand the medical problem**
    - Which is more important for this problem: sensitivity, specificity, PPV, NPV?

Robustness of saliency maps



Source: Elina Thibeau-Sutre

[Adebayo et al, 2018]

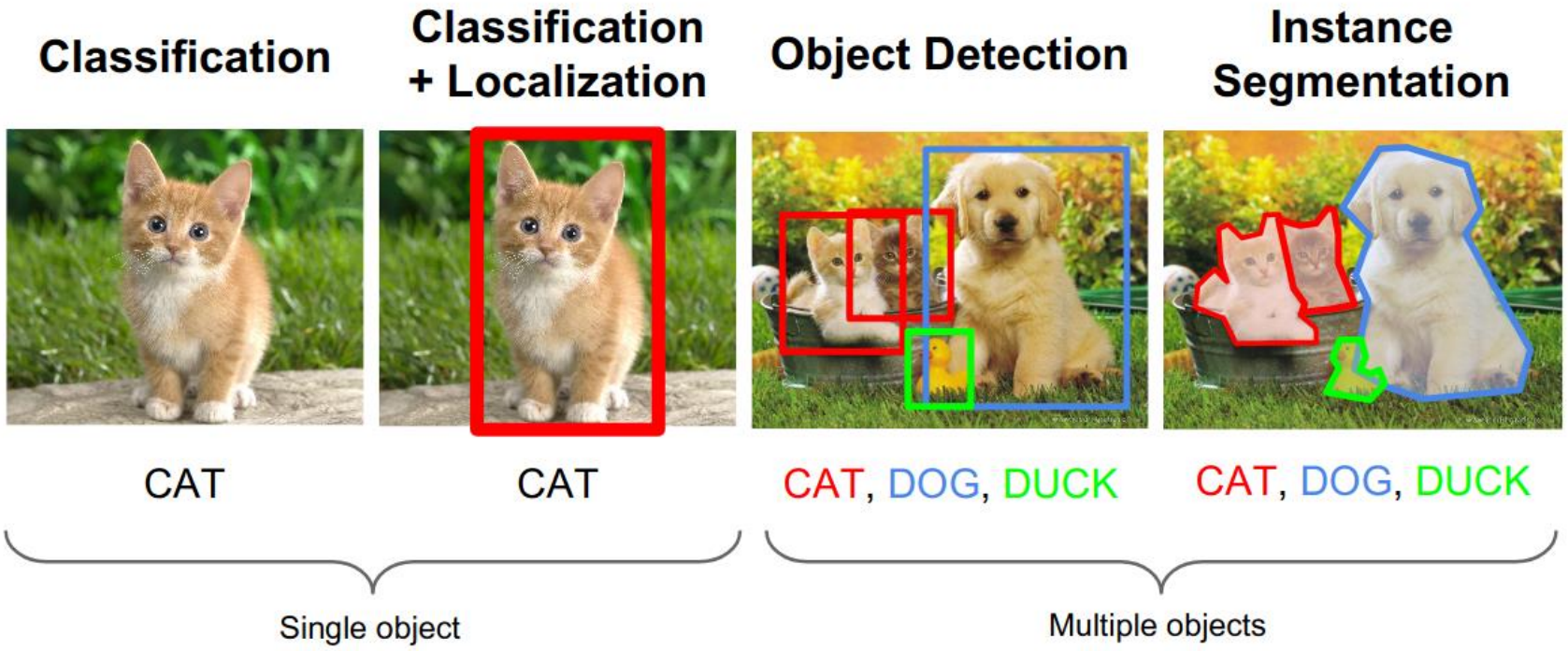
Part 4 – Detection

Object Detection

- Classification + Localization
  - Localization as Regression
  - Overfeat
- Object Detection
  - R-CNN
  - Fast-RCNN
  - Faster-RCNN
- Medical Imaging

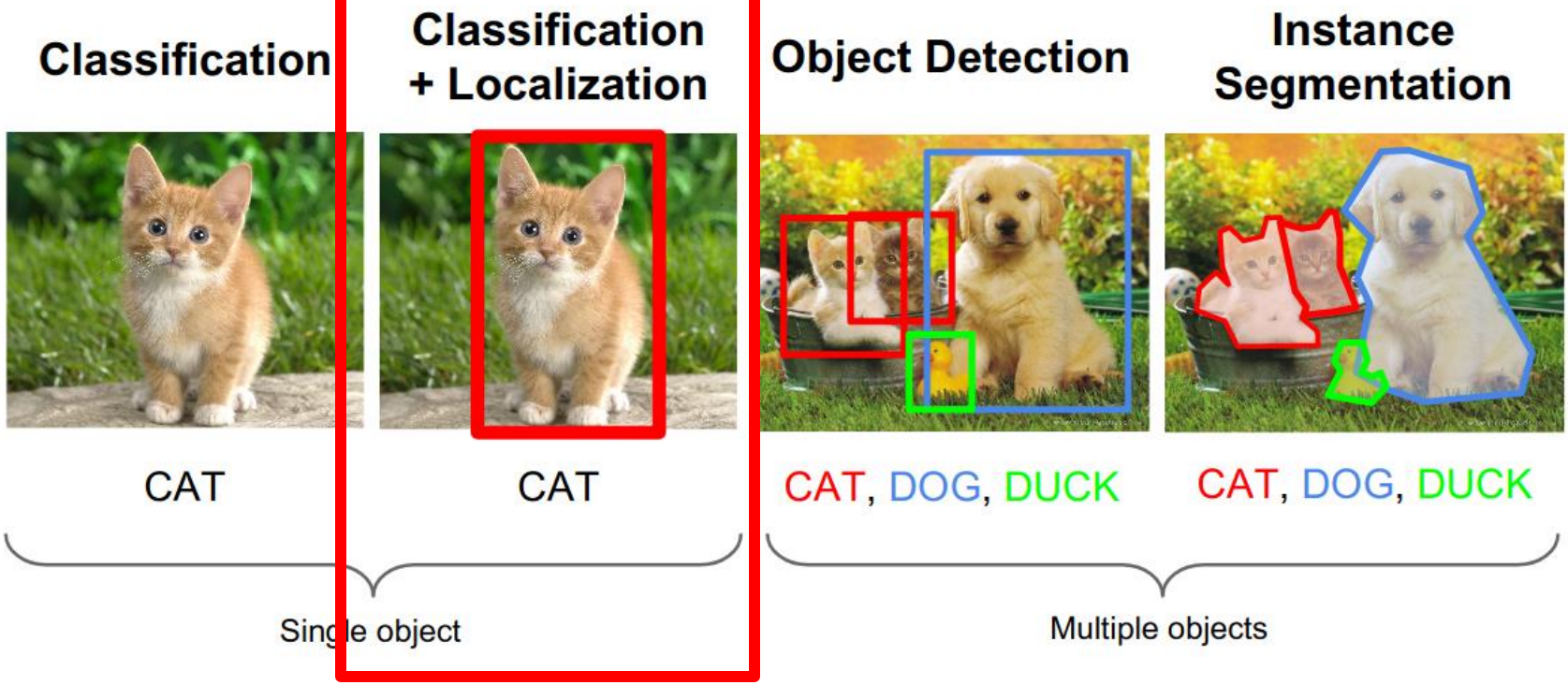
Introduction

- Different problems for vision/ similar on medical



Introduction

- Different problems for vision/ similar on medical



Introduction

- Classification + Localization: Task

- Classification: C classes
  - Input: Image
  - Output: Class label
  - Evaluation Metric: Accuracy

CAT

- Localization:
  - Input: Image
  - Output: Box in the image (x, y, w, h)
  - Evaluation Metric: Intersection over Union

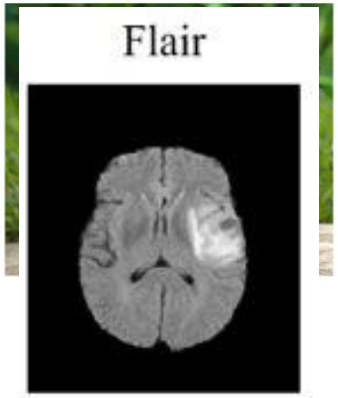
(x,y,w,h)

$$IoU = \frac{|A \cap B|}{|A \cup B|}$$

Introduction

- Classification + Localization: Task

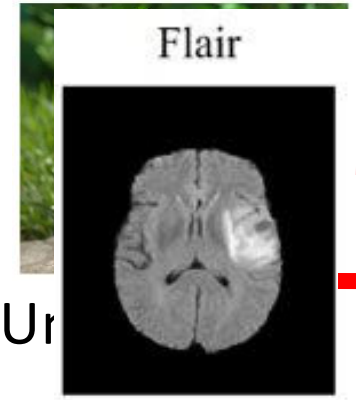
- Classification: C classes
  - Input: Image
  - Output: Class label
  - Evaluation Metric: Accuracy



CAT

Tumor

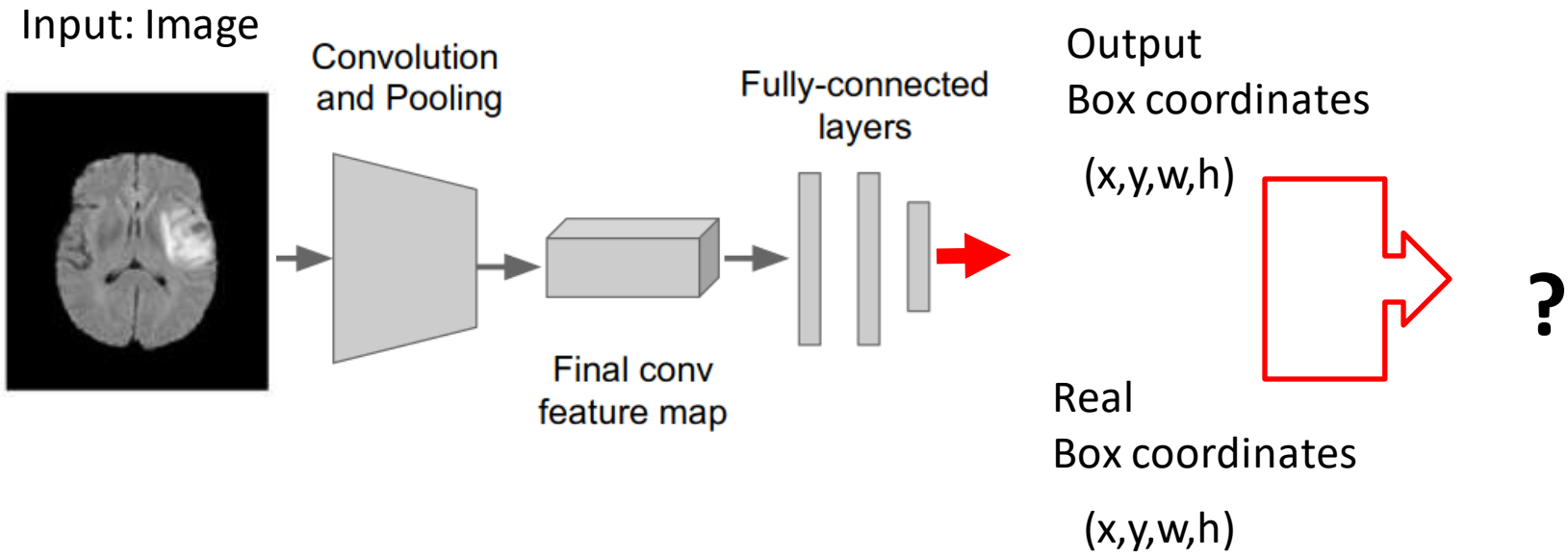
- Localization:
  - Input: Image
  - Output: Box in the image (x, y, w, h)
  - Evaluation Metric: Intersection over Ur



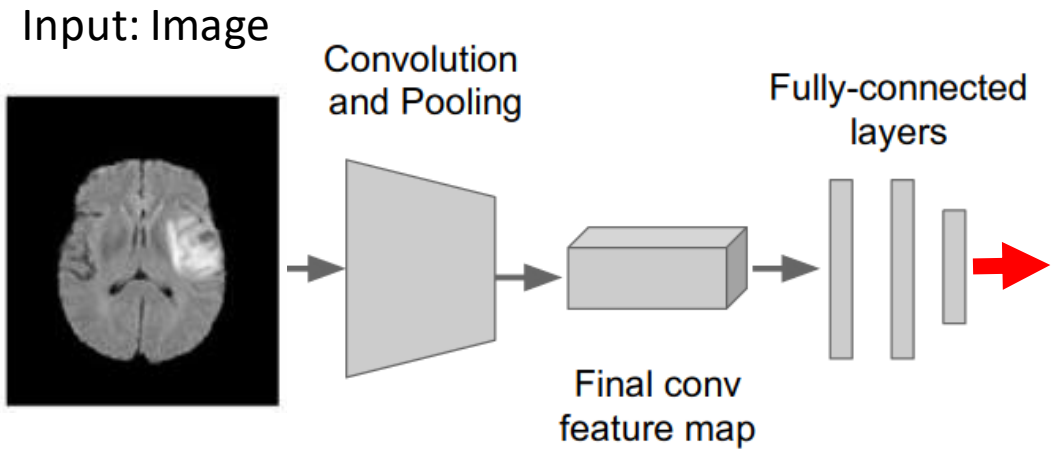
(x,y,w,h)

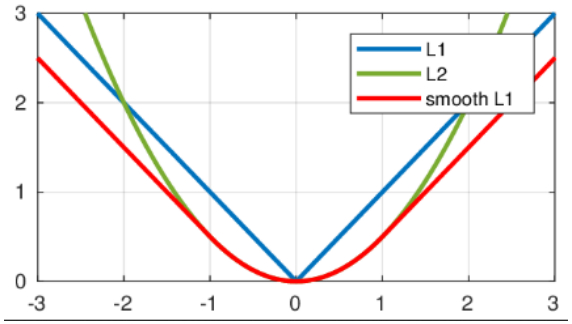
(x,y,w,h)

Localization



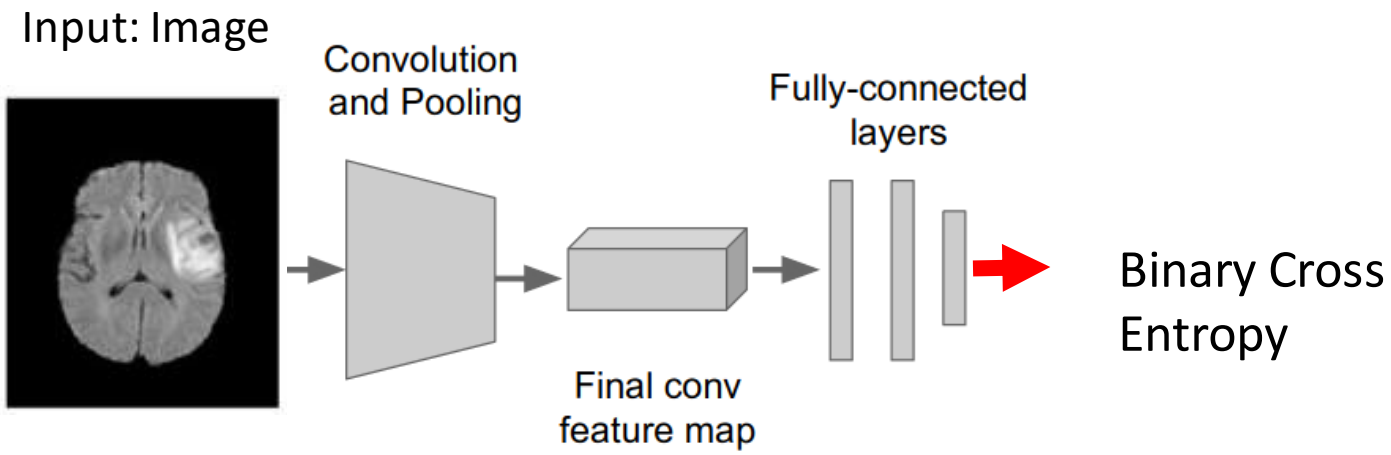
Idea: Localization as Regression





Simple Idea for Classification & Localization

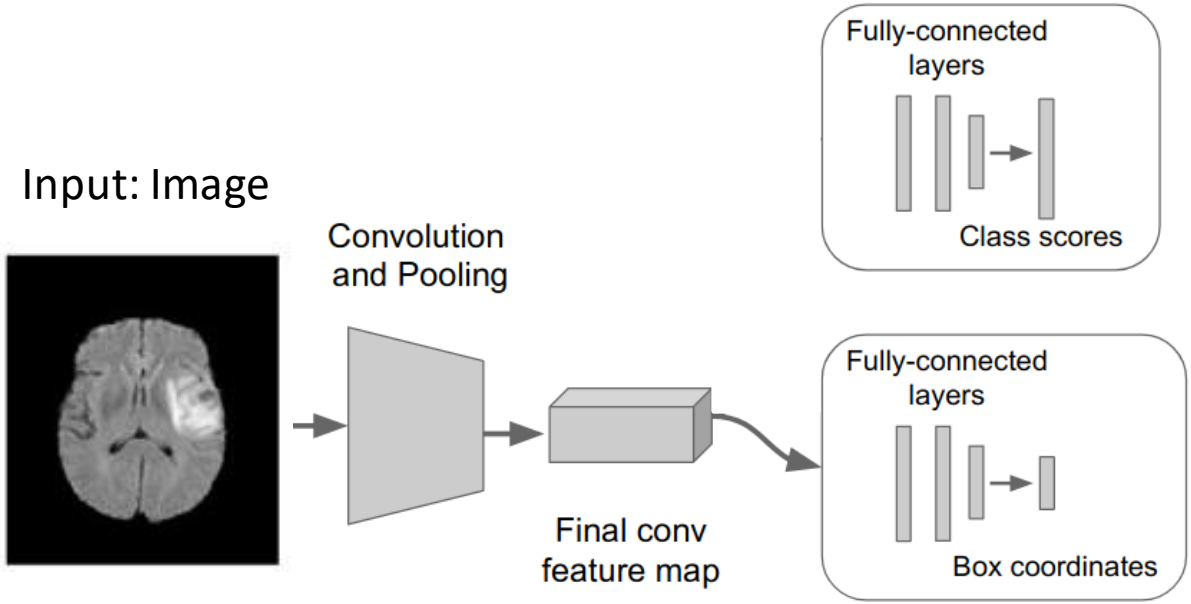
- **Step 1:** Train (or download) a classification model (VGG, ResNet, ViT...)



$$J(f) = -\frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)}\log(f(x^{(i)}) + (1-y^{(i)})log(1-f(x^{(i)}))\right)$$

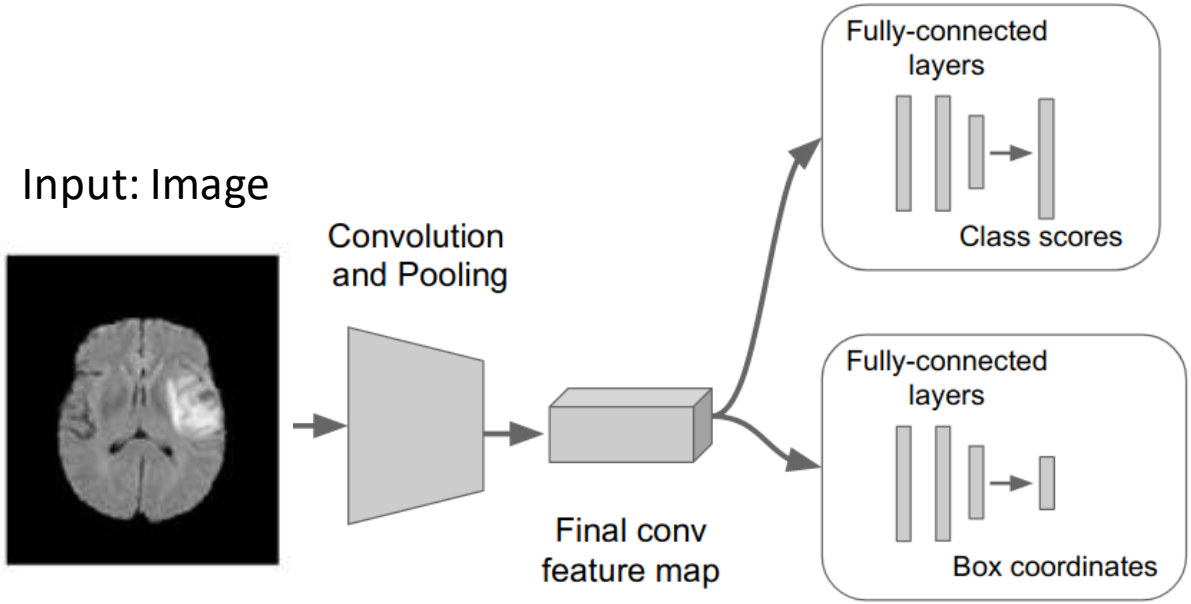
Simple Idea for Classification & Localization

- **Step 1:** Train (or download) a classification model (VGG, ResNet, ViT, ...)
- **Step 2:** Attach new fully connected "regression" to the network
- **Step 3:** Train the regression part only using the L2 loss



Simple Idea for Classification & Localization

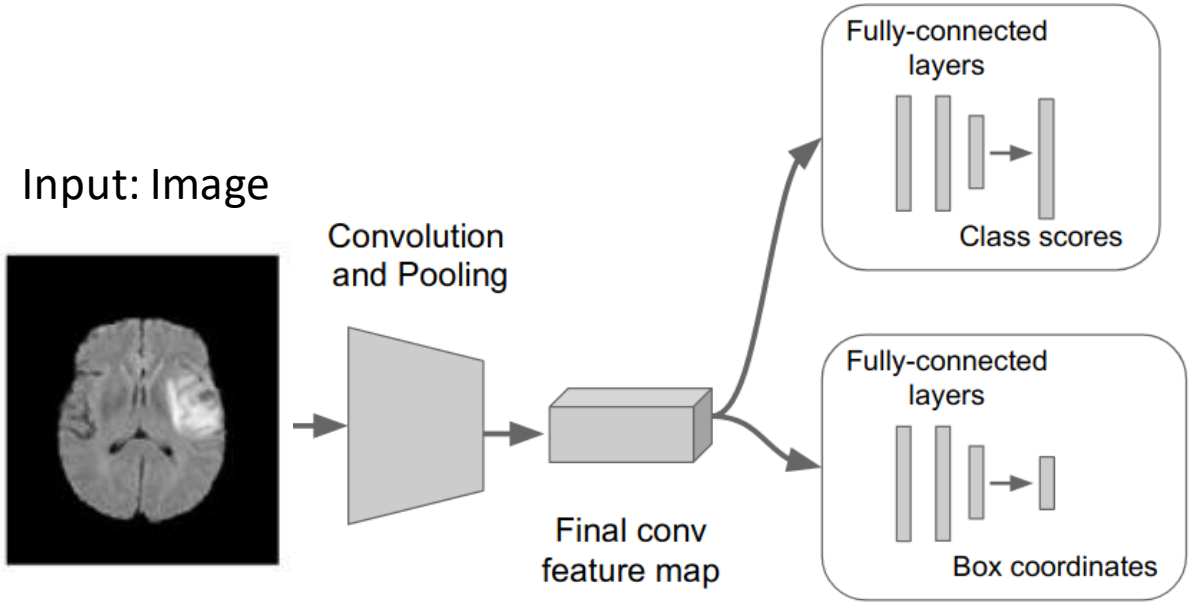
- **Step 1:** Train (or download) a classification model (VGG, ResNet, ViT ...)
- **Step 2:** Attach new fully connected "regression" to the network
- **Step 3:** Train the regression part only using the L2 loss
- **Step 4:** At test time use both branches



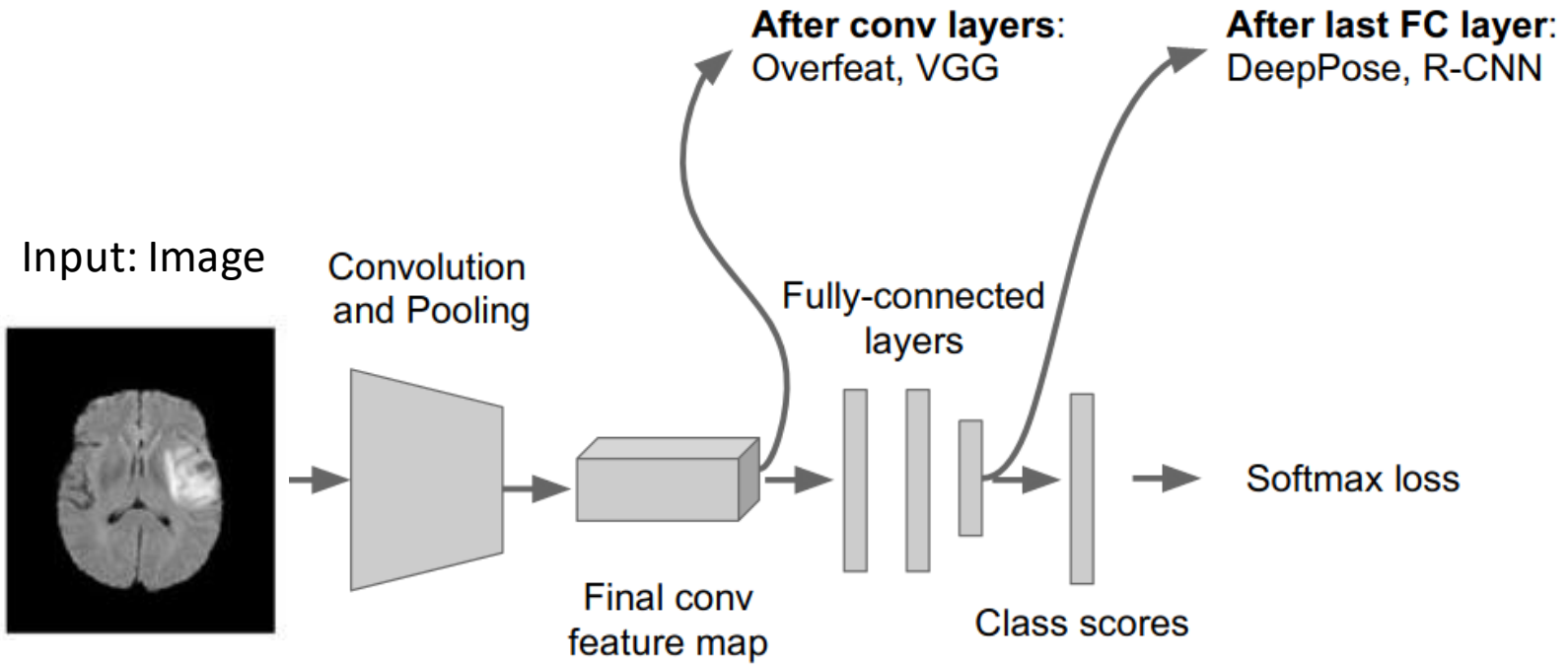
Simple Idea for Classification & Localization

- **Step 1:** Train (or download) a classification model (VGG, ResNet, ViT, ...)
- **Step 2:** Attach new fully connected "regression" to the network
- **Step 3:** Train the regression part only using the L2 loss
- **Step 4:** At test time use both branches

Possible choices???

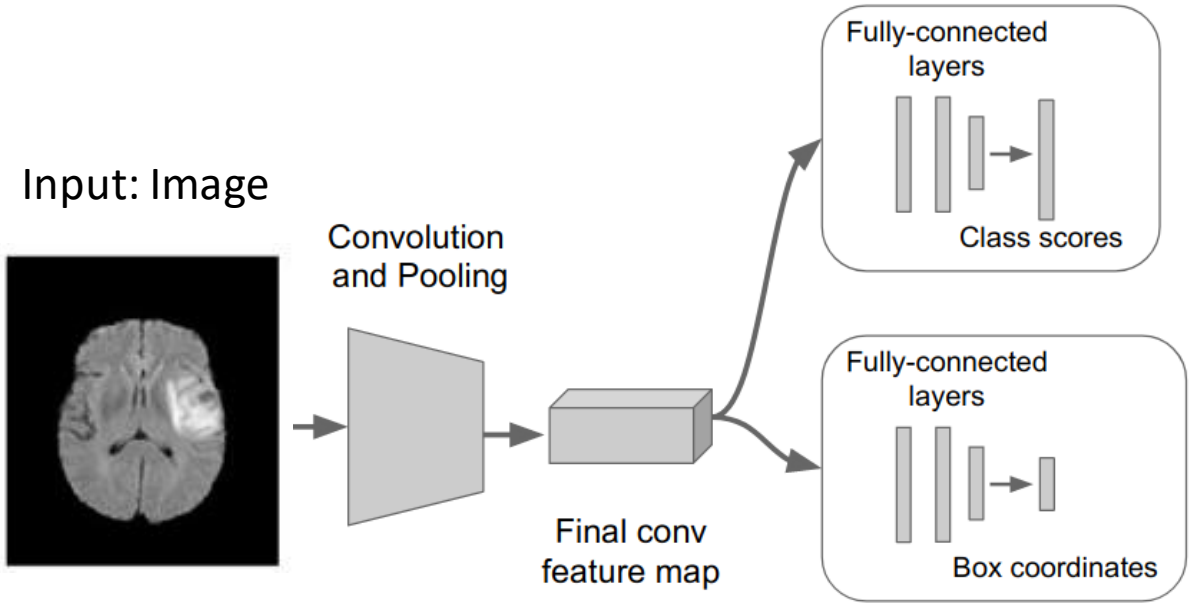


Where to attach the regression branch?



Per-class vs class agnostic regresion

- Assume we are doing classification over C classes:



Classification Branch:
C numbers (one per class)

Class agnostic:
4 numbers (One box)

Class specific
Cx4 numbers (one box per class)

Better localization?

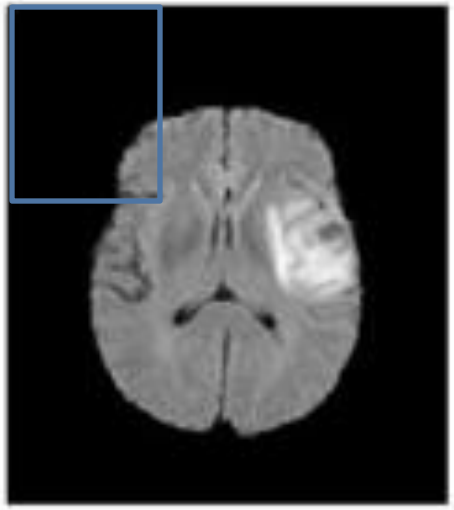
- Localization as Regression
  - Very simple
  - But ….

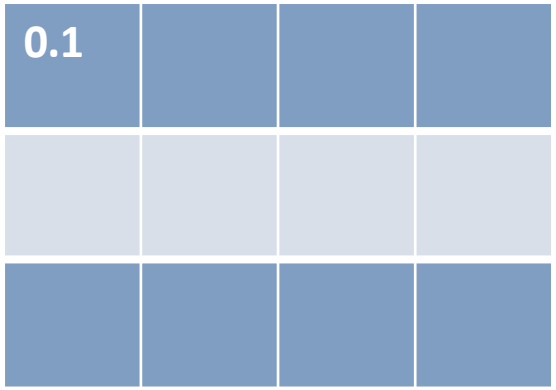
Better localization?

- Localization as Regression
  - Very simple
  - But ….

- **Idea 2: Sliding Window**
  - Run classification + localisation in multiple locations on a high-resolution image
  - Convert fully-connected layers into convolutional layers for efficient computation
  - Combine classifier and regressor predictions across all scales for final prediction

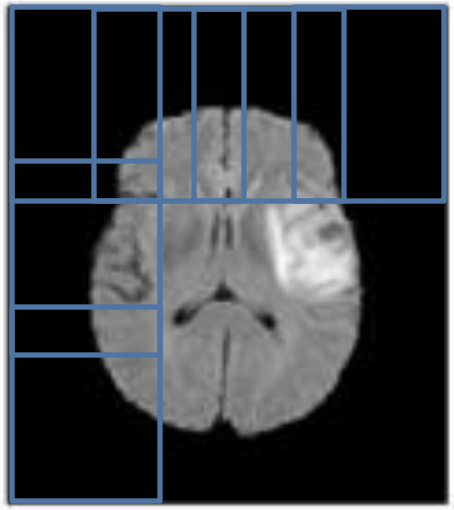
Sliding Window





Classification Scores

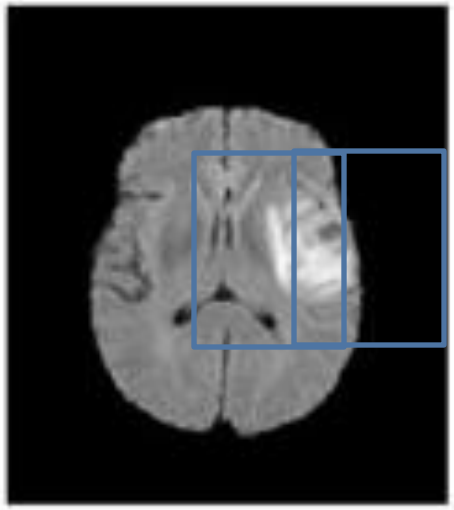
Sliding Window



0.1	0.05	0.2	0.1
0.2	0.3	0.7	0.6
0.1	0.1	0.4	0.3

Classification Scores

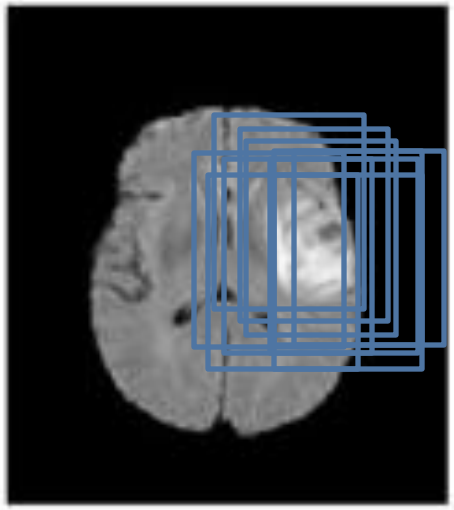
Sliding Window



0.1	**0.05**	**0.2**	**0.1**
0.2	0.3	**0.7**	**0.6**
0.1	**0.1**	**0.4**	**0.3**

Classification Scores

Sliding Window

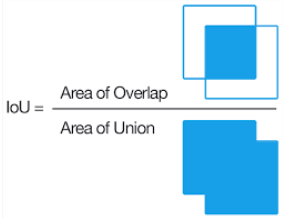


0.1	**0.05**	**0.2**	**0.1**
0.2	0.3	**0.7**	**0.6**
0.1	**0.1**	**0.4**	**0.3**

Classification Scores

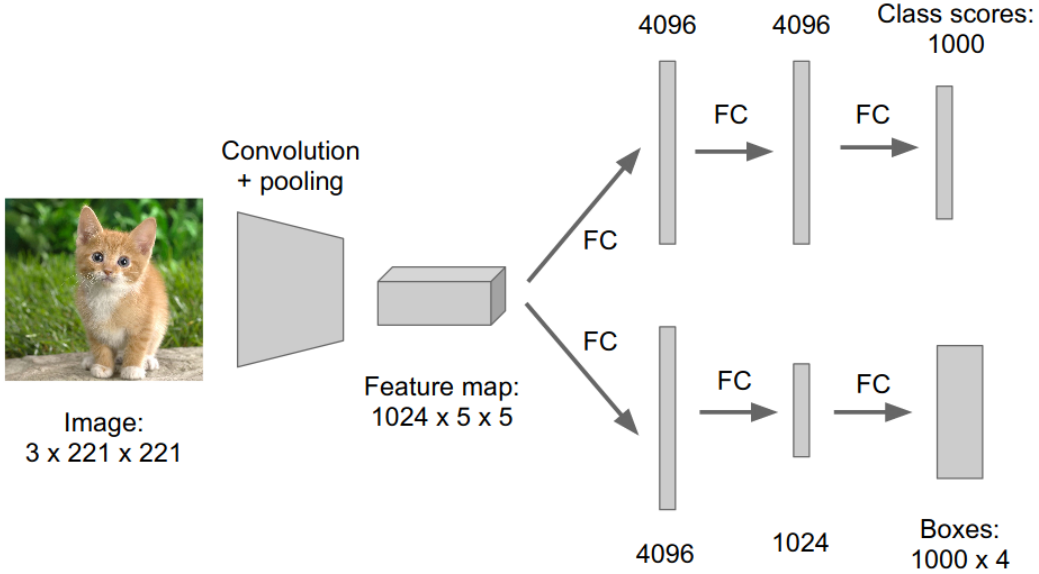
Select the proper bbox

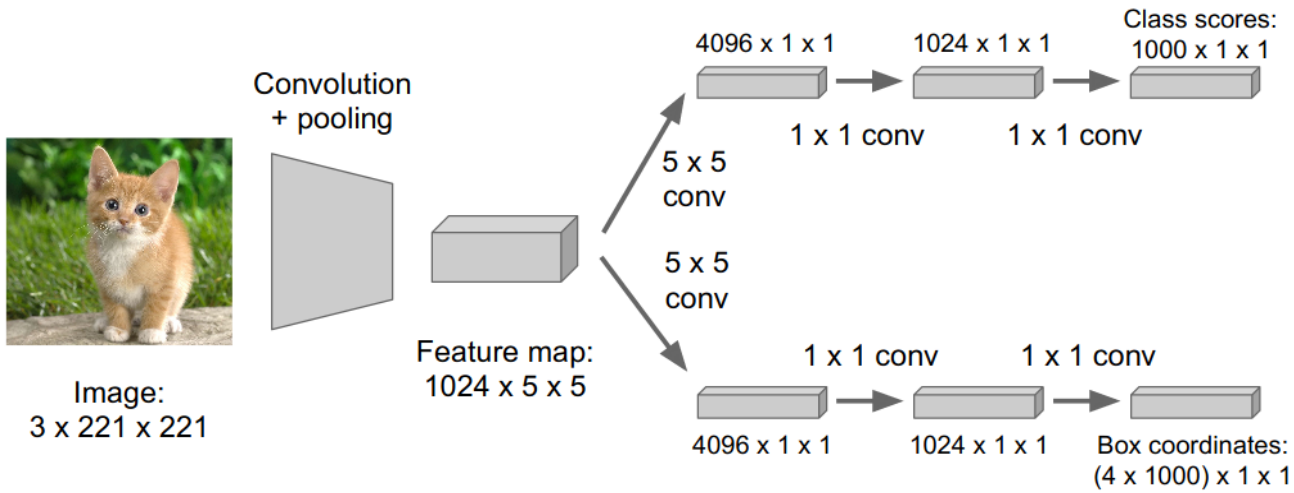
- Non-Maximum Suppression
  - Select the bboxes with the highest probability
  - Calculate their intersection and disregard bbxes with IoU> thrs.



Sliding Window - Overfeat

- Winner of ILSVRC 2013 Localization challenge



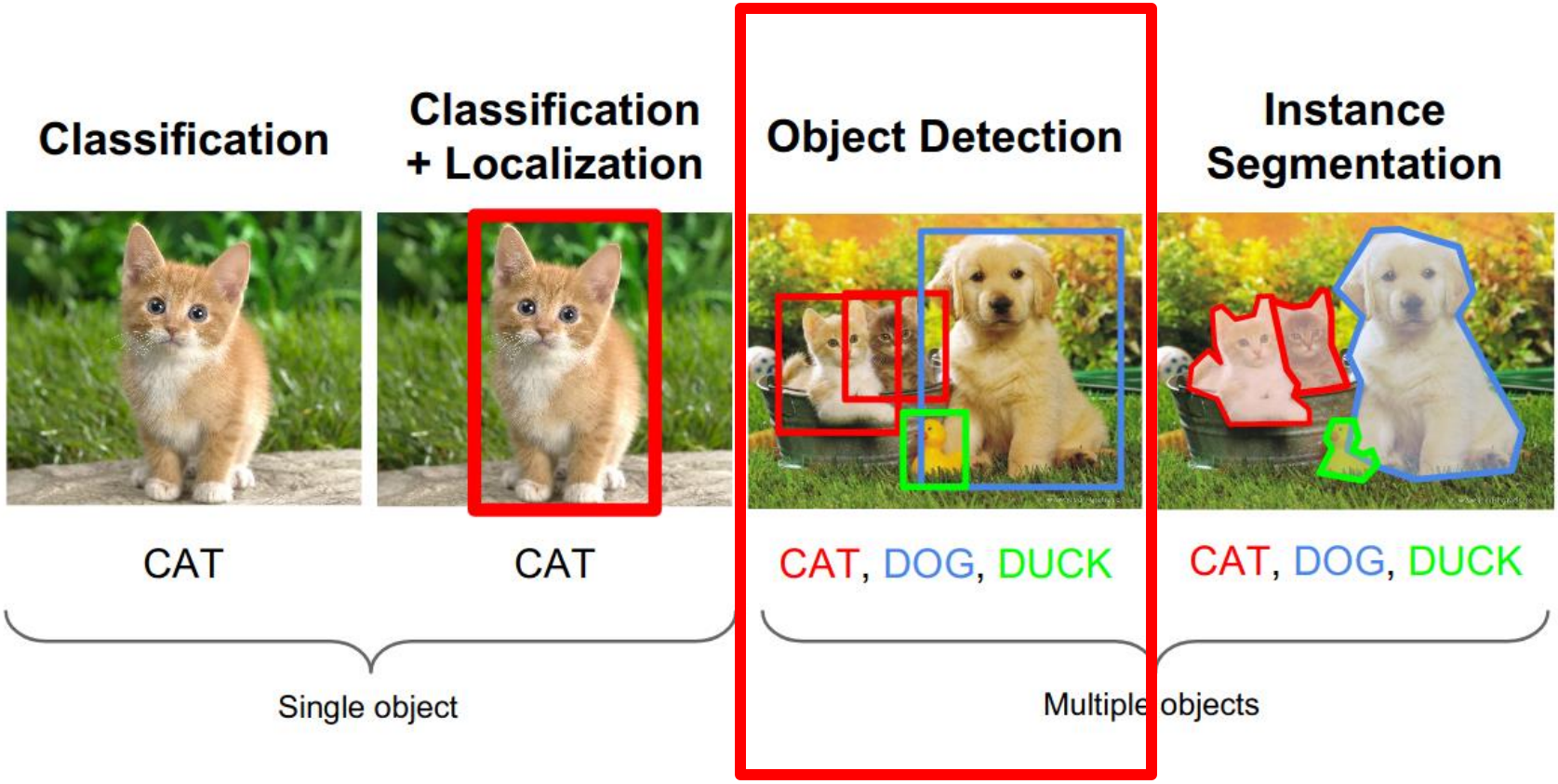


Localization & Classification

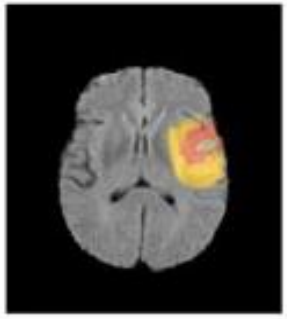
Recap

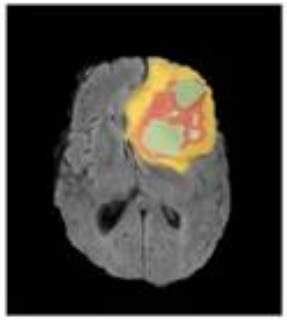
- Find a fixed number of objects (one or many)
- L2 regression from CNN features to box coordinates
- Overfeat: Regression + efficient sliding window with FC -> conv conversion
- Deeper networks do better

Object detection



Object detection as Regression?





- GD-enhance tumor (x,y,w,h)
- Peritumoral edema (x,y,w,h)
- Non-enhancing tumor core (x,y,w,h)

12 parameters to regress

- GD-enhance tumor (x,y,w,h)
- Peritumoral edema (x,y,w,h)
- Non-enhancing tumor core (x,y,w,h)
- Non-enhancing tumor core (x,y,w,h)

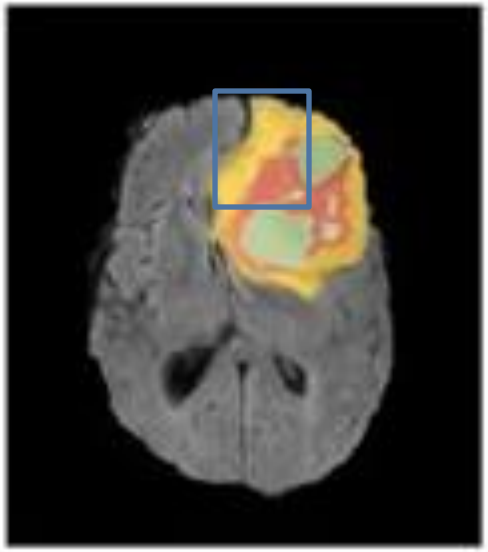
16 parameters to regress

. . .

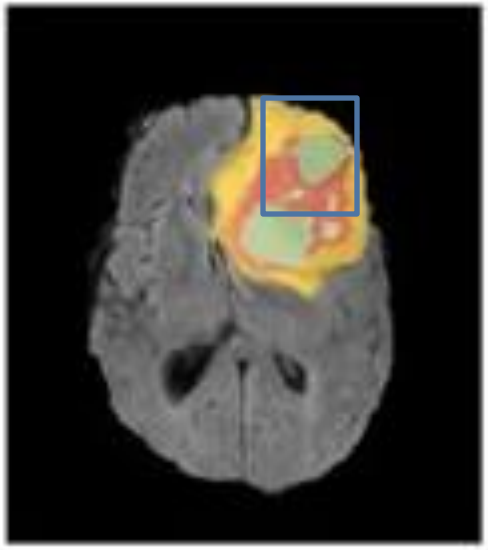
. . .

Need of variable sized outputs

Object detection as Classification?



GD-enhance tumor ? No
Peritumoral edema ? No
Non-enhancing tumor core? No



GD-enhance tumor ? No
Peritumoral edema ? No
Non-enhancing tumor core? Yes

- **Problem:** ????

Object detection as Classification?

- **Problem:** Need to test many positions and scales
  - Apply CNN to every possible crop of the image and take the class

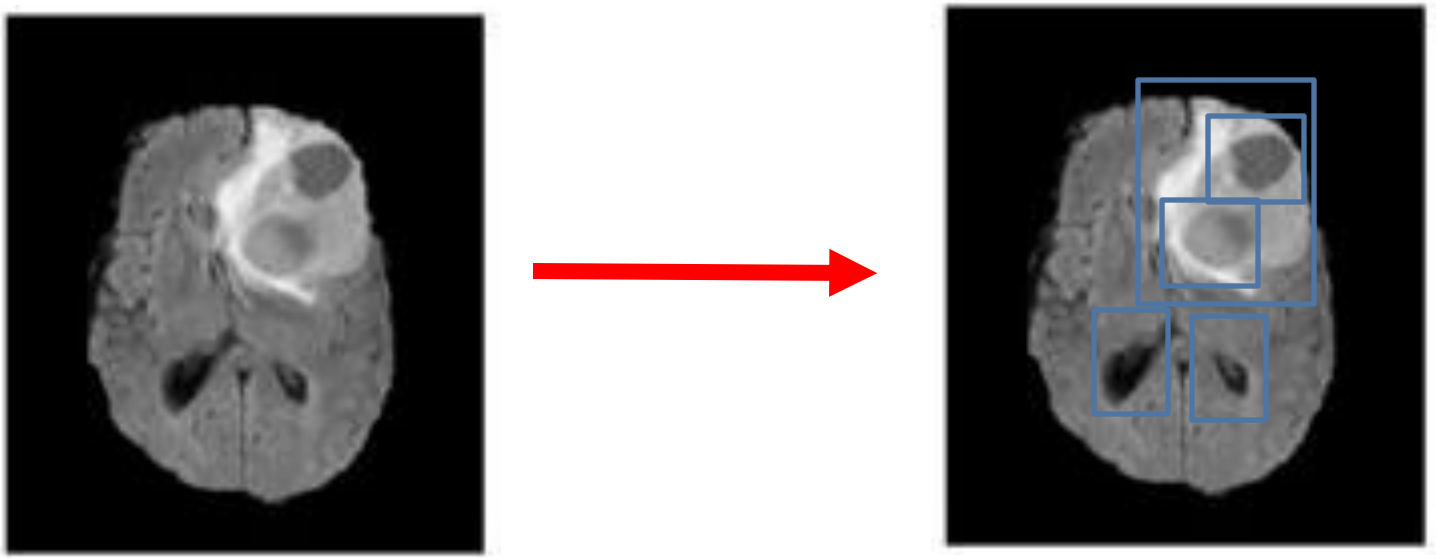
- **Solution 1:** If your classifier is fast enough, just do it

Object detection as Classification?

- **Problem:** Need to test many positions and scales
  - Apply CNN to every possible crops of the image and take the class
- **Solution 1:** If your classifier is fast enough, just do it
- **Solution 2:** Only look at a tiny subset of possible positions
  - Find image regions that are likely to contain objects (e.g. interensting image areas)

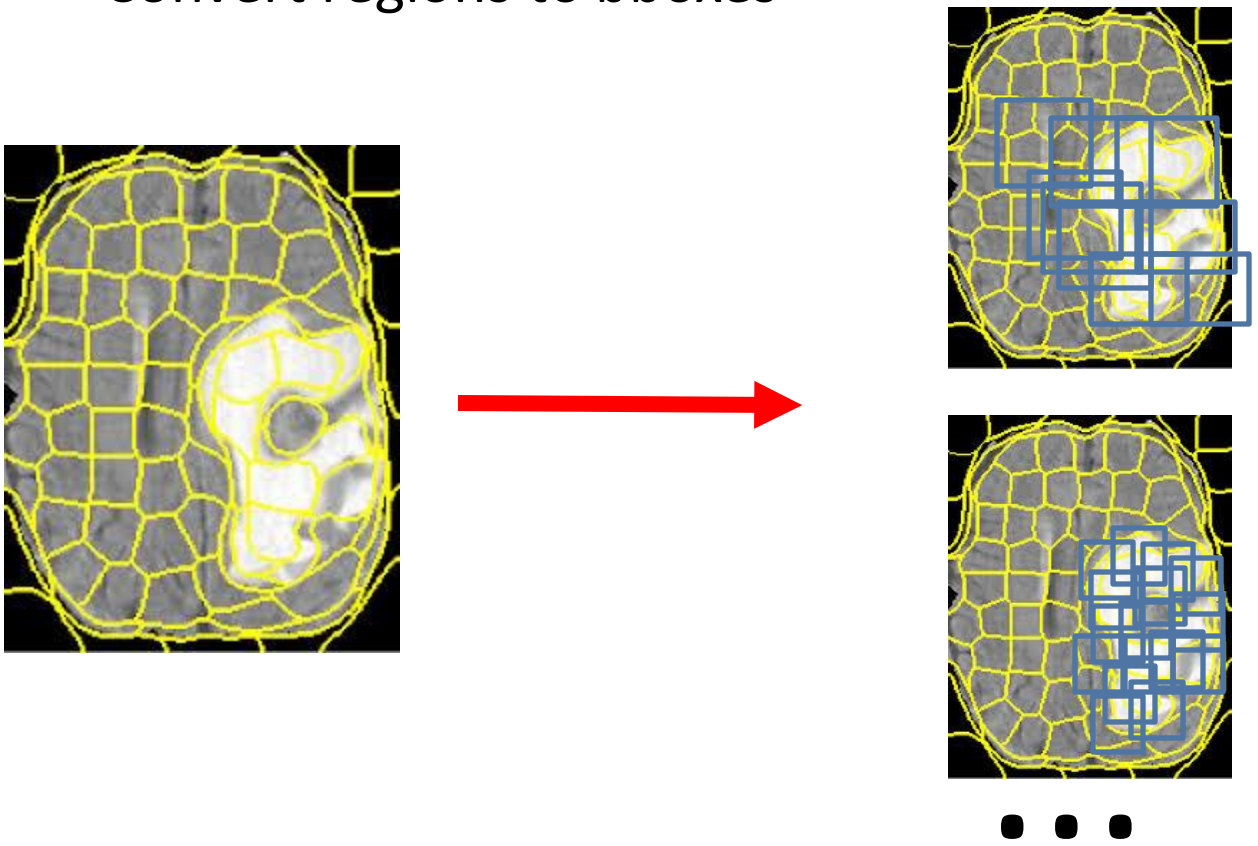
Region Proposals

- Find image regions that are likely to contain objects
- Class independant object detection
- Look for blob-like regions



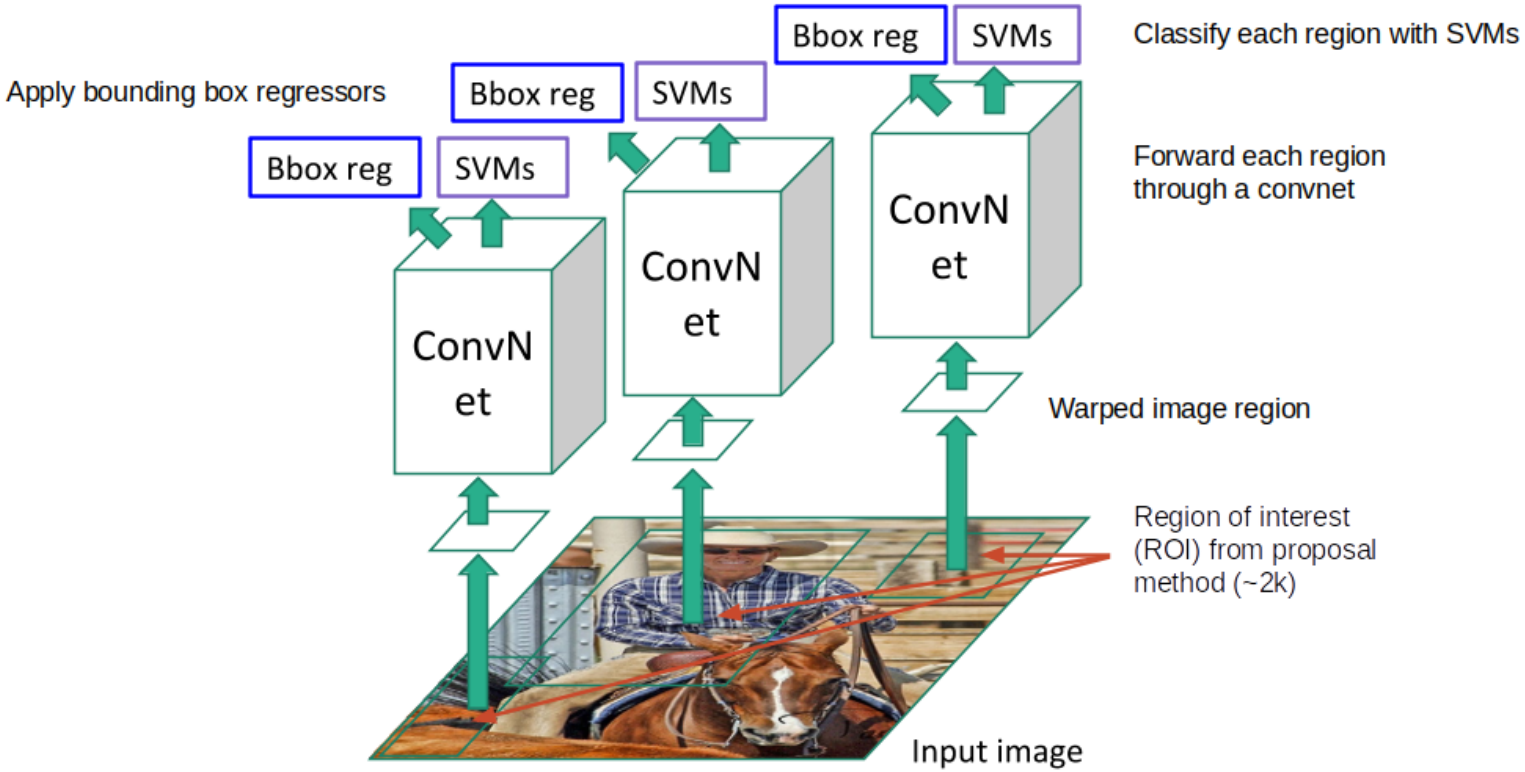
Region Proposals

- **Selective Search:** Bottom-up segmentation, merging regions at multiple scales.
  - Convert regions to bboxes



Ujilings et al, "Selective Search for Object Recognition", IJCV 2013

R-CNN



R. Girshick et al. CVPR 2014

Region Proposals

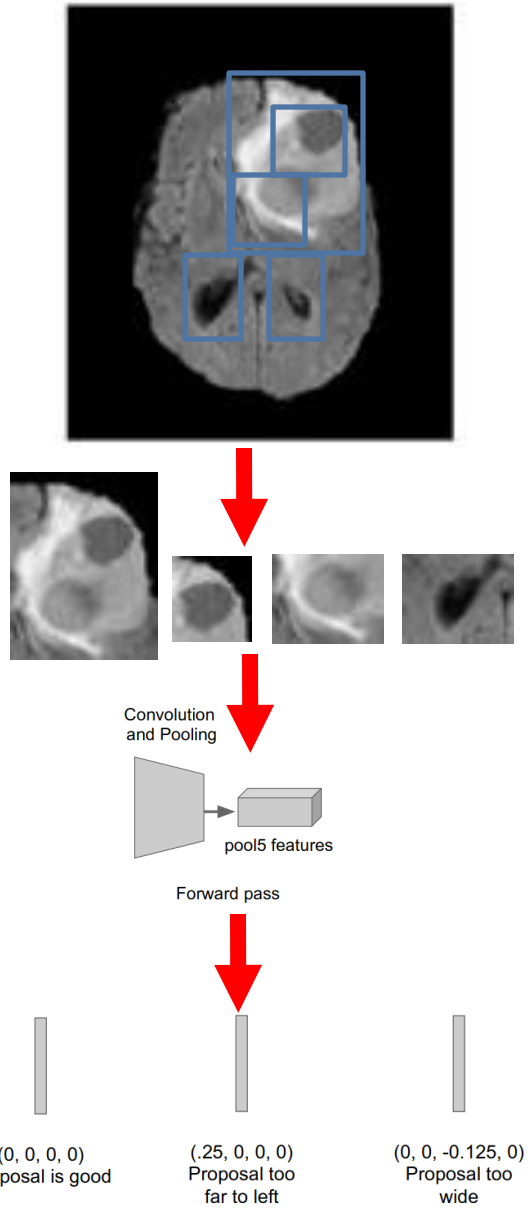
- Many many choices ….

Method	Approach	Outputs Segments	Outputs Score	Control #proposals	Time (sec.)	Repea-tability	Recall Results	Detection Results
Bing [18]	Window scoring		✓	✓	0.2	⋆⋆⋆	⋆	·
CPMC [19]	Grouping	✓	✓	✓	250	-	⋆⋆	⋆
EdgeBoxes [20]	Window scoring		✓	✓	0.3	⋆⋆	⋆⋆⋆	⋆⋆⋆
Endres [21]	Grouping	✓	✓	✓	100	-	⋆⋆⋆	⋆⋆
Geodesic [22]	Grouping	✓		✓	1	⋆	⋆⋆⋆	⋆⋆
MCG [23]	Grouping	✓	✓	✓	30	⋆	⋆⋆⋆	⋆⋆⋆
Objectness [24]	Window scoring		✓	✓	3	·	⋆	·
Rahtu [25]	Window scoring		✓	✓	3	·	·	⋆
RandomizedPrim's [26]	Grouping	✓		✓	1	⋆	⋆	⋆⋆
Rantalankila [27]	Grouping	✓		✓	10	⋆⋆	·	⋆⋆
Rigor [28]	Grouping	✓		✓	10	⋆	⋆⋆	⋆⋆
SelectiveSearch [29]	Grouping	✓	✓	✓	10	⋆⋆	⋆⋆⋆	⋆⋆⋆
Gaussian				✓	0	·	·	⋆
SlidingWindow				✓	0	⋆⋆⋆	·	·
Superpixels		✓			1	⋆	·	·
Uniform				✓	0	·	·	·

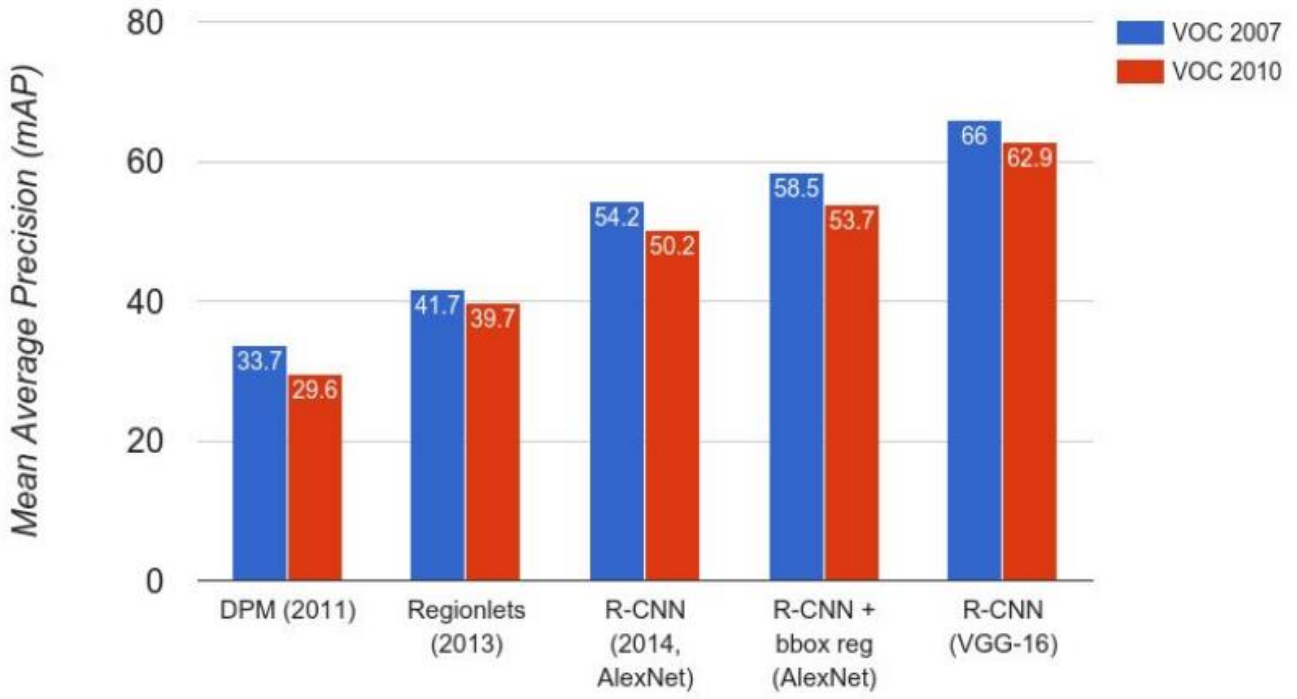
Hosang et al., "What makes for effective detection proposals?", PAMI 2015

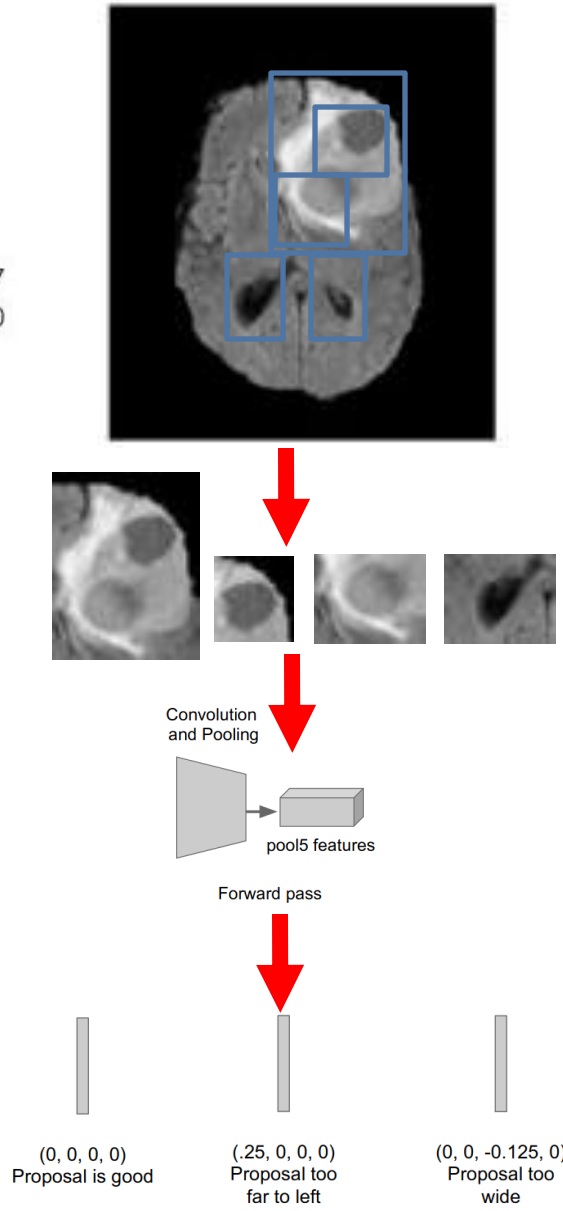
R-CNN – Recipe

- Step 1: Train a classification model for your medical problem (or use a pretrained one)
- Step 2: Fine-tune model for detection
  - Train for the number of object classes you have on your dataset!
- Step 3: Extract features
  - Extract region proposals for all images
  - For each region: warp to CNN input size, run forward through CNN, save the features of the last layer
- Step 4: Train one binary SVM per class to classify region features
- Step 5: For each class, train a linear regression model to map from cached features to offsets to GT boxes to make up for "slightly wrong" proposals



R-CNN – Results

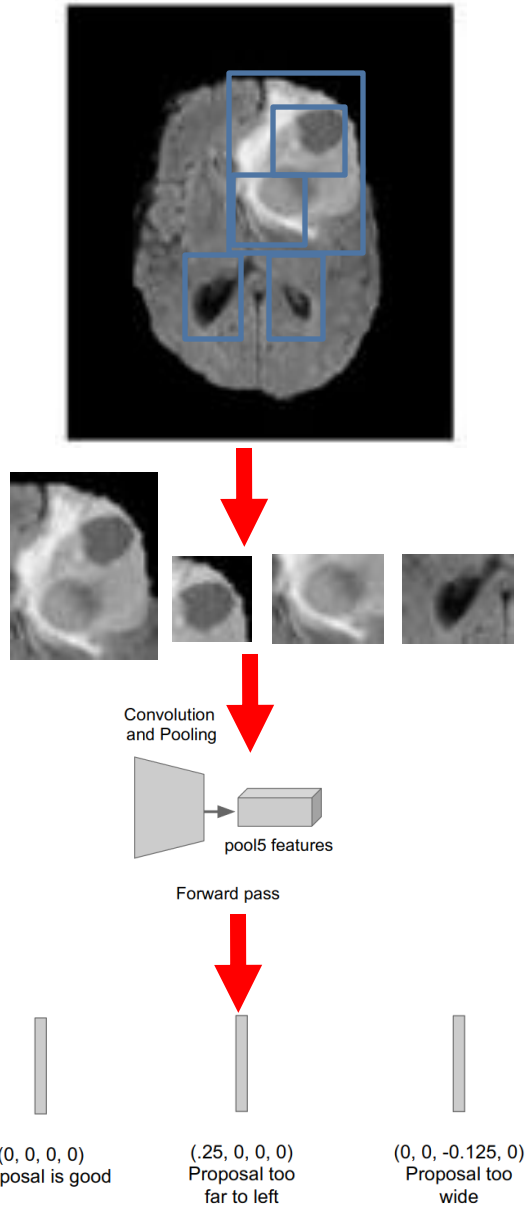




Wang et al, ICCV 2013

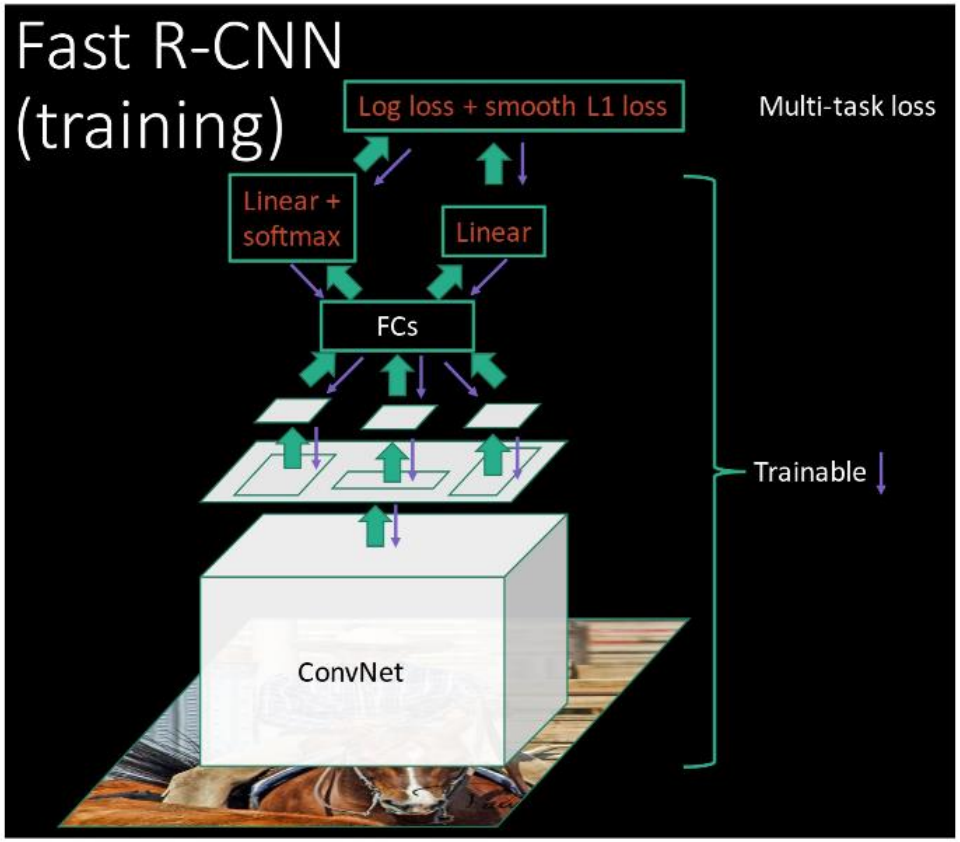
R-CNN – Problems

- Ad-hoc training objectives
  - Fine-tune network with softmax classifier (log loss)
  - Train post-hoc linear SVMs (hinge loss)
  - Train post-hoc bounding-box regressors (L2 loss)
- Training is slow, takes a lot of disk space
- Inference is slow
- Complex multistage training pipeline



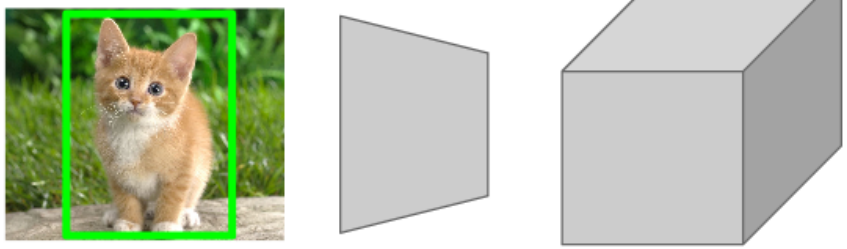
Fast R-CNN

- Train the whole system end-to-end all at once!
- Share computation of convolutional layers between proposals for an image.



R. Girshick et al. ICCV 2015

Fast R-CNN: Region of Interest Pooling

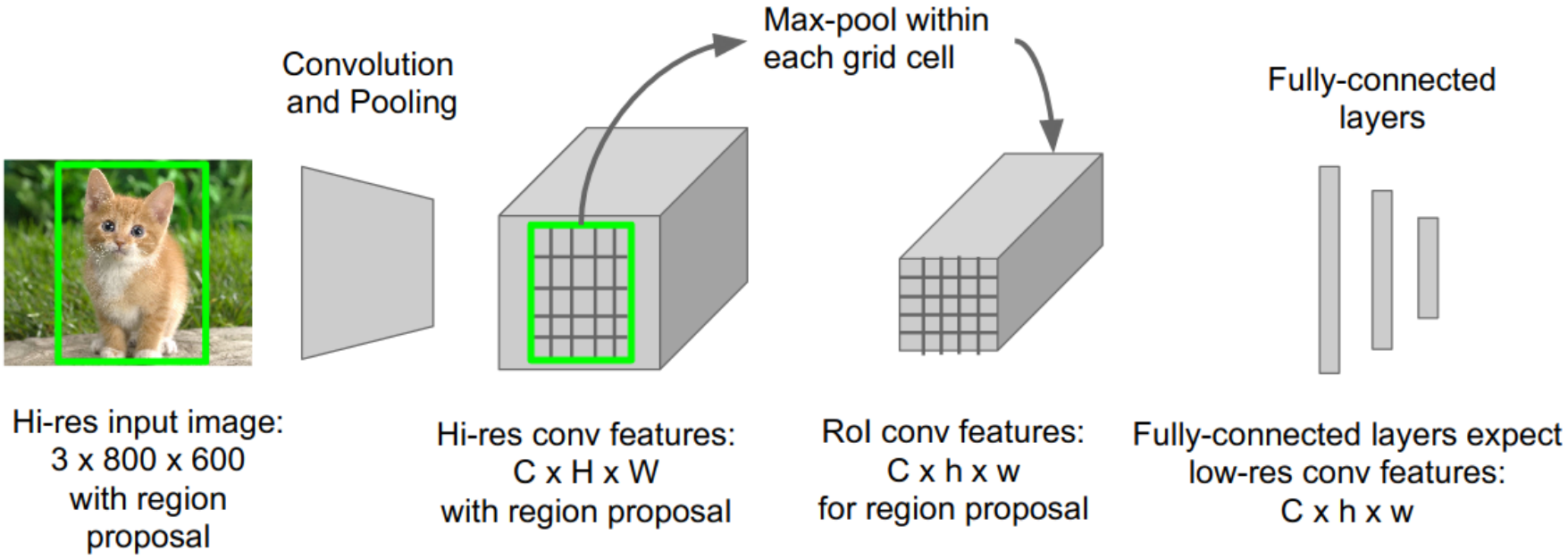


Hi-res input image:
3 x 800 x 600
with region
proposal

Hi-res conv features:
C x H x W
with region proposal

Problem: Fully-connected layers expect low-res conv features: C x h x w

Fast R-CNN: Region of Interest Pooling



- Train is enabled as all the process are differentiable!

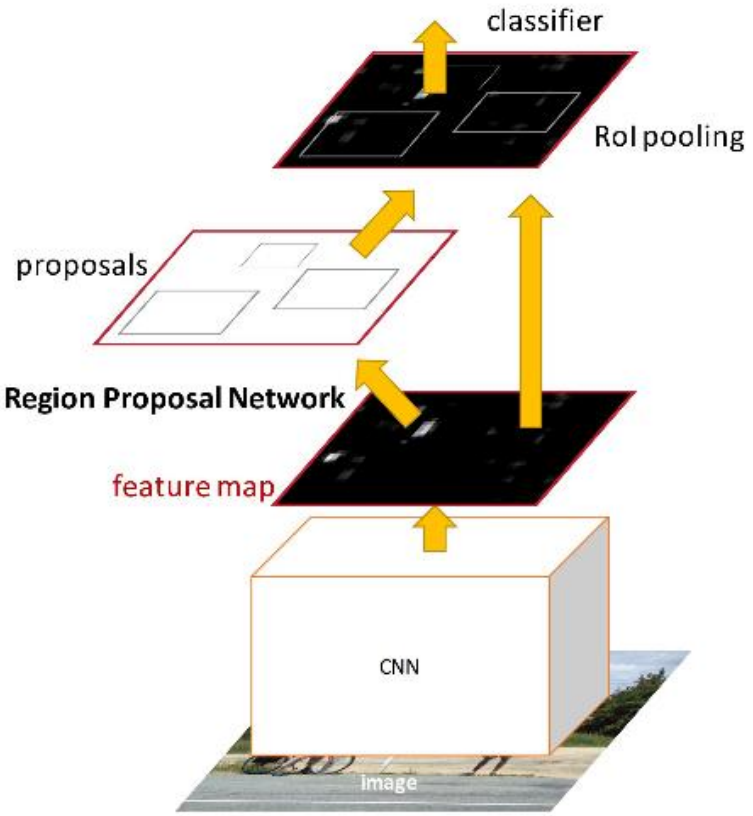
R-CNN vs Fast R-CNN

	R-CNN	Fast R-CNN
Faster!	Training Time:	84 hours	**9.5 hours**
	(Speedup)	1x	**8.8x**
FASTER!	Test time per image	47 seconds	**0.32 seconds**
	(Speedup)	1x	**146x**
Better!	mAP (VOC 2007)	66.0	**66.9**

Using VGG-16 CNN on Pascal VOC 2007 dataset

- However, the region proposals are still a separate step in the process! Slowing the process

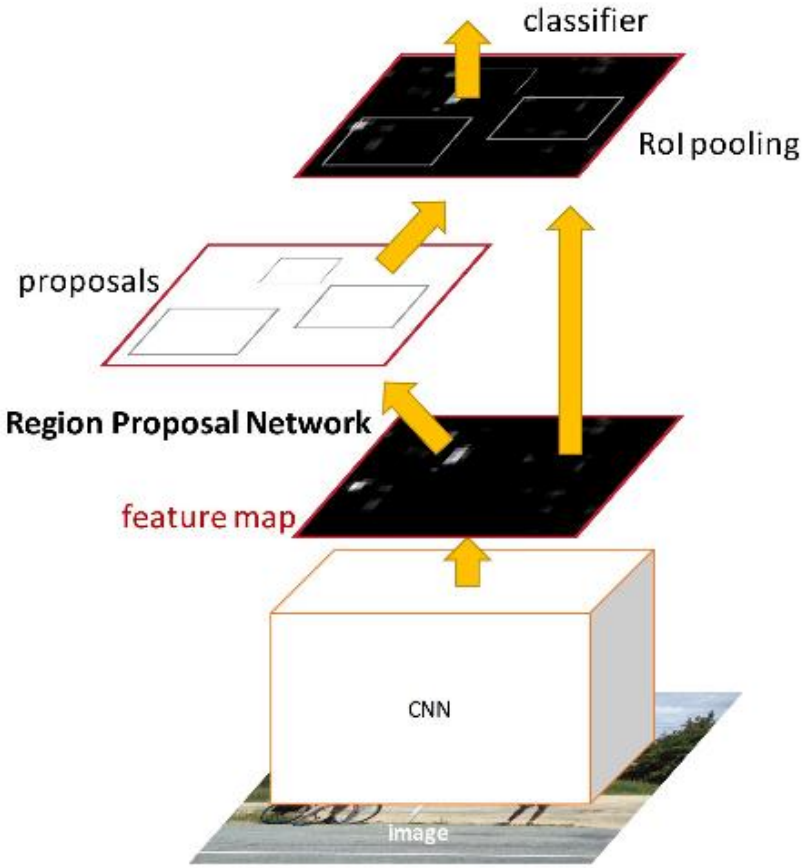
Faster R-CNN



- Insert a **Region Proposal Network (RPN)** after the last convolutional layer.
- After RPN,use RoI Pooling and an upstream classifier and bbox regressor just like Fast R-CNN.

Ren et al. NeurIPS 2015

Faster R-CNN: Region Proposal Network



Ren et al. NeurIPS 2015

Slide a small window on the feature map

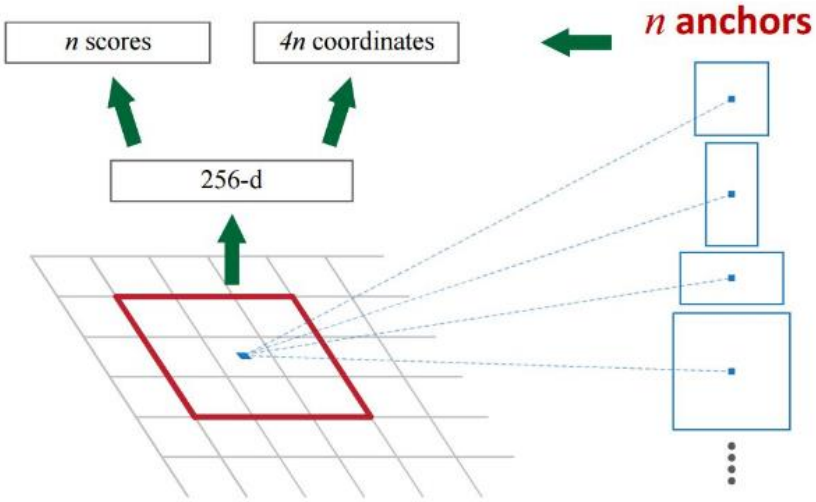
Build a small network for:

- Classifying object or not-object,
- Regressing bbox locations

Position of the sliding window provides localization information with reference to the image

Box regression provides finer localization information with reference to this sliding window.

Faster R-CNN: Region Proposal Network



Use N anchor boxes at each location

Anchors are translation invariant: use the same ones at every location

Regression gives offsets from anchor boxes

Classification gives the probability that each (regressed) anchor shows an object.

Ren et al. NeurIPS 2015

R-CNN vs Fast R-CNN vs Faster R-CNN

	R-CNN	**Fast R-CNN**	**Faster R-CNN**
Test time per image (with proposals)	50 seconds	2 seconds	**0.2 seconds**
(Speedup)	1x	25x	**250x**
mAP (VOC 2007)	66.0	**66.9**	**66.9**

Ren et al. NeurIPS 2015

Object Detection choices ...

- Base Network
  - VGG16
  - ResNet ...
  - Inception …
  - MobileNet
  - ViTs
  - …
- Object Detection architecture
  - Faster R-CNN
  - R-FCN
  - SSD
  - DETR
  - …

More networks for Object Detection

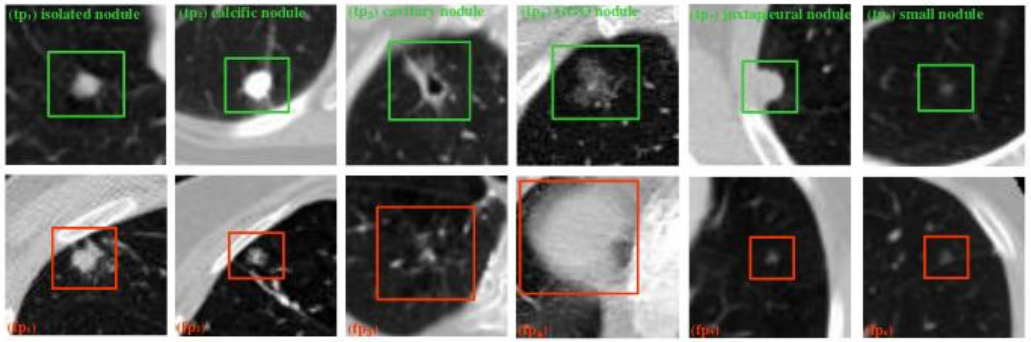
- Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection", arXiv:1506.02640
- Johnson et al., “DenseCap: Fully Convolutional Localization Networks for Dense Captioning”, CVPR 2016
- Lin et al., "Focal Loss for Dense Object Detection", ICCV 2017
- Berman M. et al., "The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks" CVPR 2018
- ….

Good but what about medical imaging?

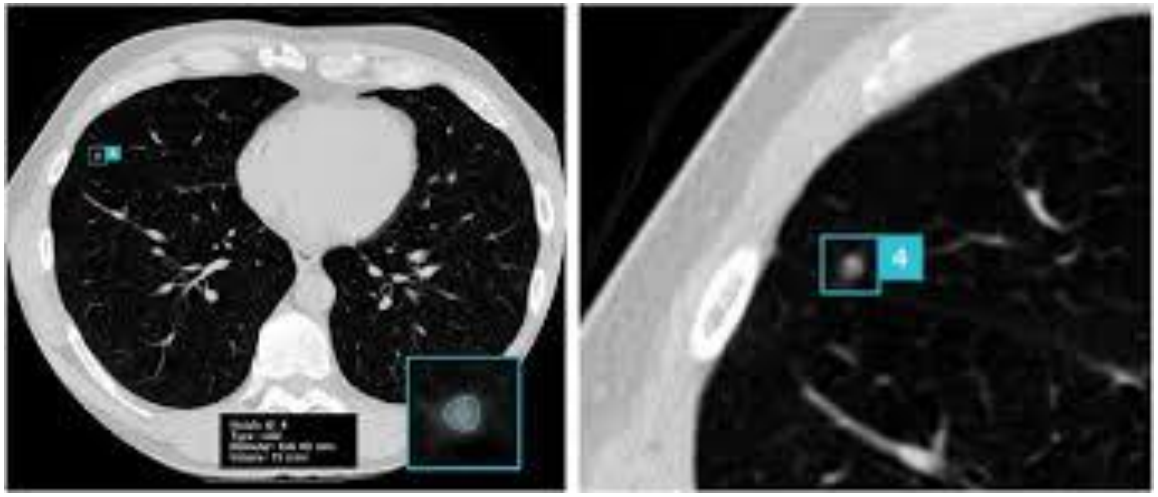
- Problems?

Good but what about medical imaging?

- Problems?
  - Extreme Class Imbalance Objects are very small and rare
  - The separation of the object of interest from the background is not easy.



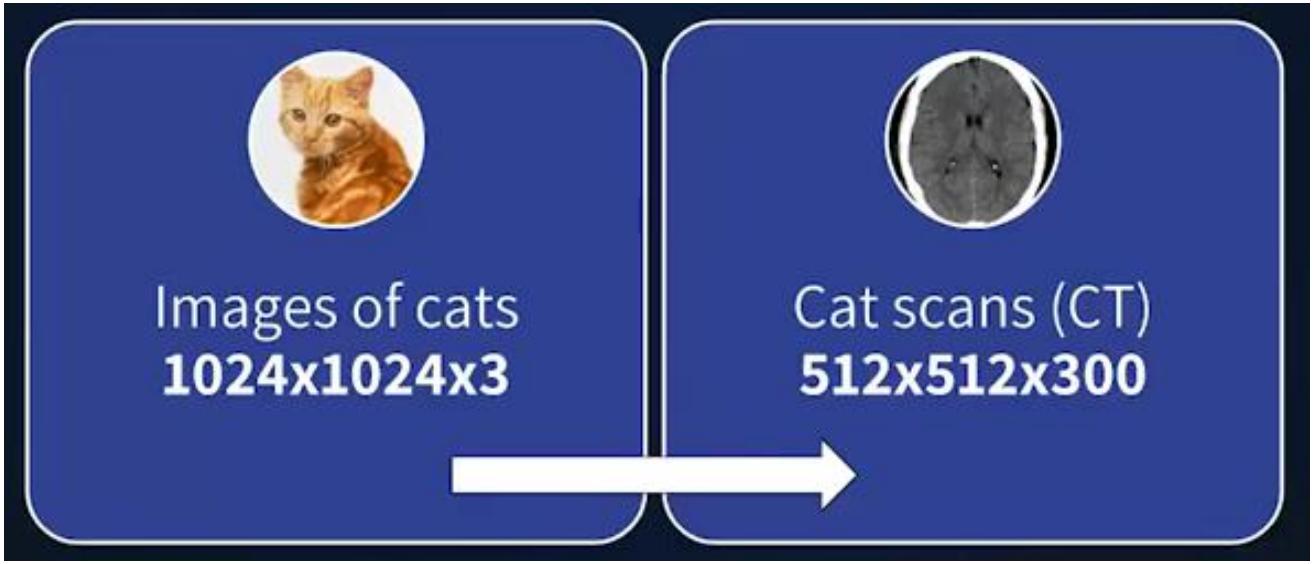
Cao et al., ArXiv 2019



Perez et al., SIPAIM 2017

Good but what about medical imaging?

- Problems?
  - Extreme Class Imbalance Objects are very small and rare
  - The separation of the object of interest from the background is not easy.
  - Detection is needed in a Large 3D Volume

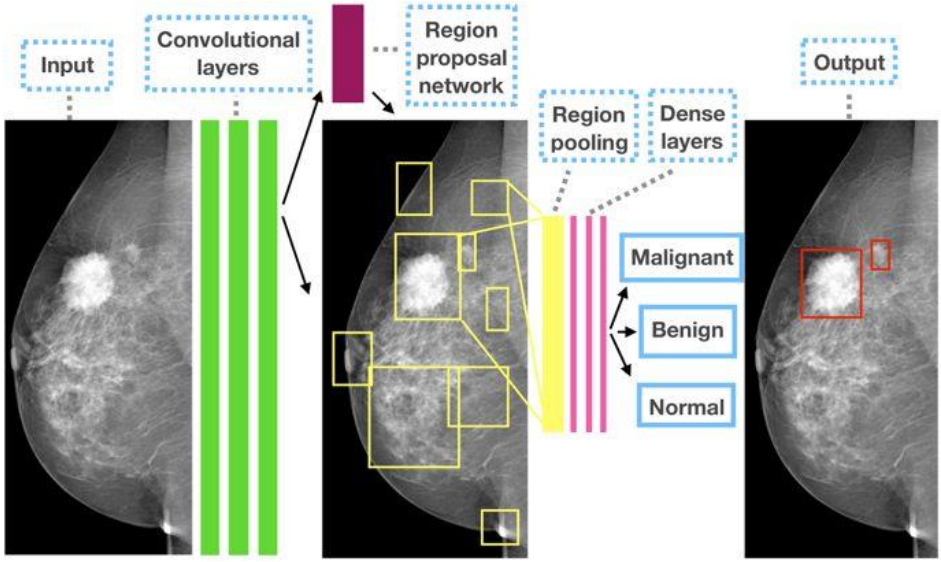


Good but what about medical imaging?

- Problems?
  - Extreme Class Imbalance Objects are very small and rare
  - The separation of the object of interest from the background is not easy.
  - Detection is needed in a Large 3D Volume
  - Detection will be followed by clinical relevant questions that will need to integrate more data!

Detecting lesions in mammograms

- Use Faster-RCNN with VGG-16
- Classification on malignant or benign lesions on a mammogram
- AUC: 0.95
- Use of publicly available dataset including (MIAS)



MENU SCIENTIFIC REPORTS

Article | Open Access | Published: 15 March 2018

Detecting and classifying lesions in mammograms with Deep Learning

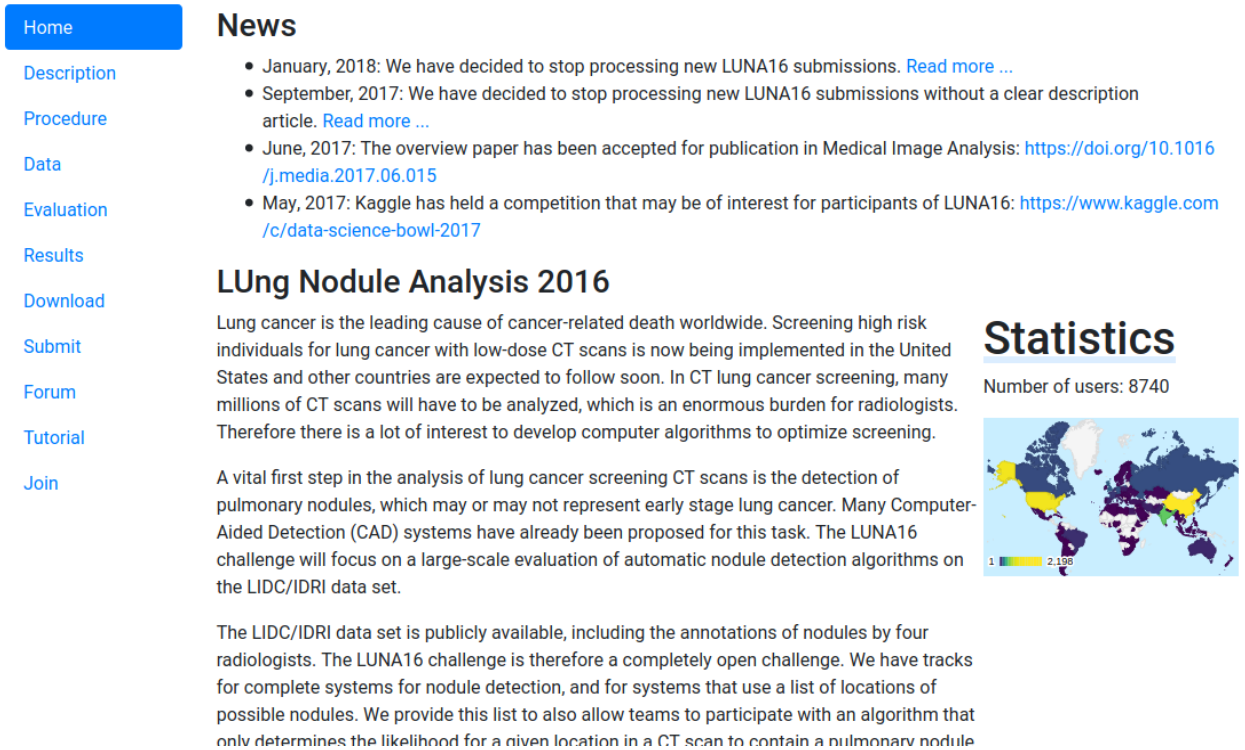
Dezső Ribli, Anna Horváth, Zsuzsa Unger, Péter Pollner & István Csabai

Scientific Reports **8**, Article number: 4165 (2018) | Cite this article

Nodule detection

LUNA Dataset

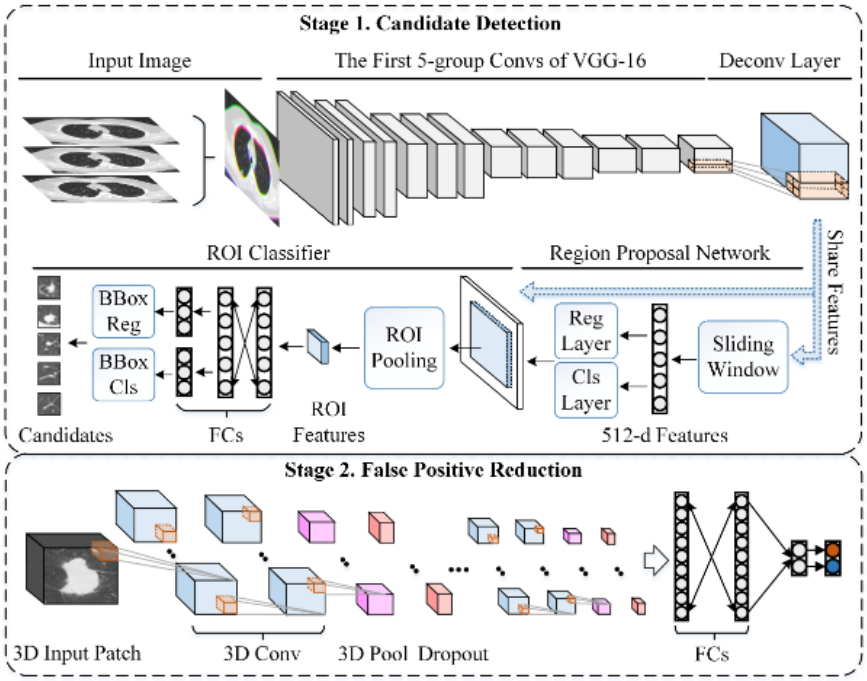
- A big number of false positives for nodule detection due to very small size
- 2 stage process!



Nodule detection

LUNA Dataset

- A big number of false positives for nodule detection due to very small size
- 2 stage process!
- Faster R-CNN to detect the bboxes of possible nodules
- 3D network to decide if the detection is good or not!



International Conference on Medical Image Computing and Computer-Assisted Intervention
MICCAI 2017: Medical Image Computing and Computer Assisted Intervention – MICCAI 2017 pp 559-567 | Cite as

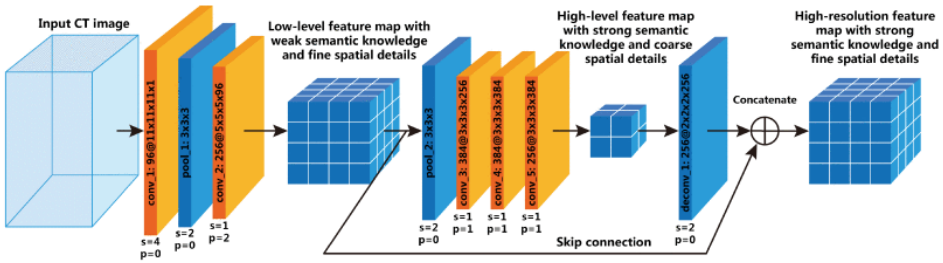
Accurate Pulmonary Nodule Detection in Computed Tomography Images Using Deep Convolutional Neural Networks

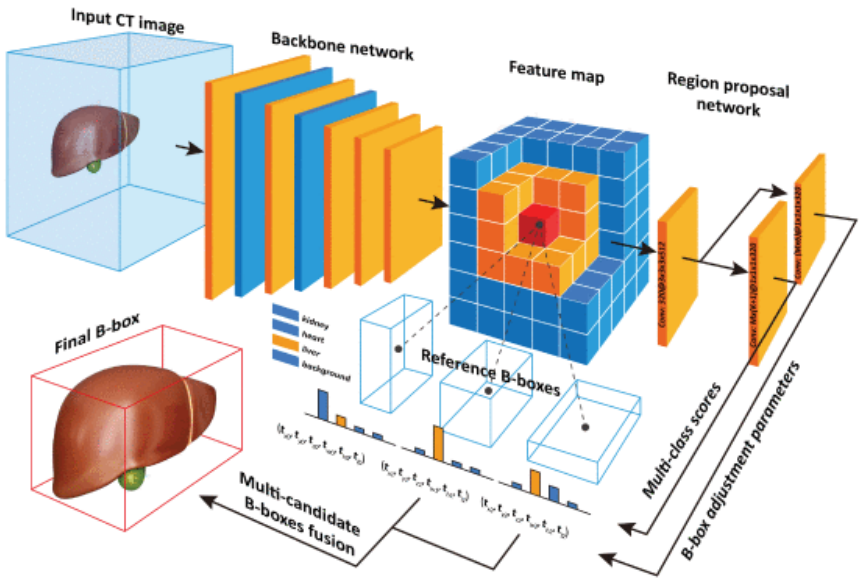
Authors
Authors and affiliations
Jia Ding, Aoxue Li, Zhiqiang Hu, Liwei Wang

Conference paper
First Online: 04 September 2017
40 Citations
9.3k Downloads
Part of the Lecture Notes in Computer Science book series (LNCS, volume 10435)

Multiple Organ Localization in CT

- 3D version of Faster R-CNN.
- Use of 3D convolutions





Journals & Magazines > IEEE Transactions on Medical ... > Volume: 38 Issue: 8

Efficient Multiple Organ Localization in CT Image Using 3D Region Proposal Network

Datasets Available

Publisher: IEEE

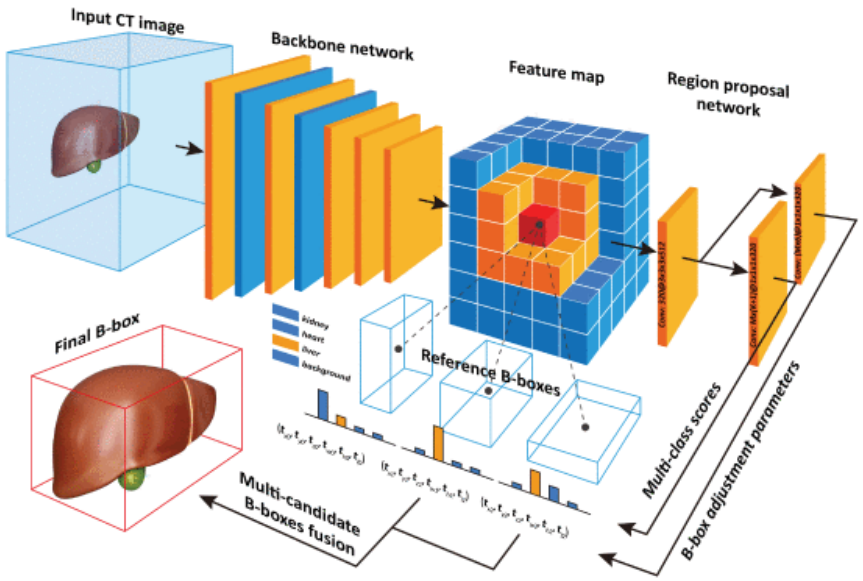
5 Author(s) Xuanang Xu ; Fugen Zhou ; Bo Liu ; Dongshan Fu ; Xiangzhi Bai **View All Authors**

1 Paper Citation

1554 Full Text Views

Multiple Organ Localization in CT

- 3D version of Faster R-CNN.
- Use of 3D convolutions
- Region Proposal Network, similar to the Faster R-CNN, regressiong 6 parameters (t_x,t_y,t_z,t_w,t_h,t_l)



Journals & Magazines > IEEE Transactions on Medical ... > Volume: 38 Issue: 8

Efficient Multiple Organ Localization in CT Image Using 3D Region Proposal Network

Datasets Available

Publisher: IEEE

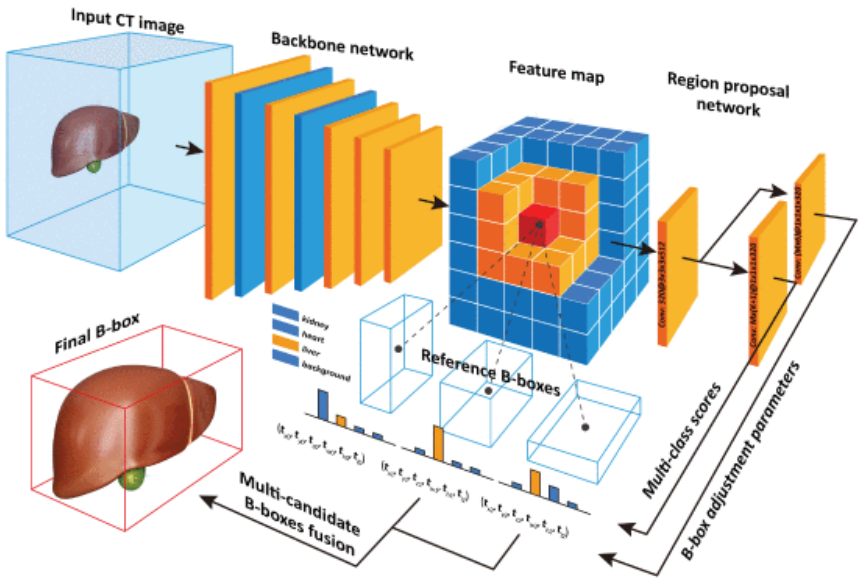
5 Author(s) Xuanang Xu ; Fugen Zhou ; Bo Liu ; Dongshan Fu ; Xiangzhi Bai **View All Authors**

1 Paper Citation

1554 Full Text Views

Multiple Organ Localization in CT

- 3D version of Faster R-CNN.
- Use of 3D convolutions
- Region Proposal Network, similar to the Faster R-CNN, regressiong 6 parameters (t_x,t_y,t_z,t_w,t_h,t_l)
- Use of focal loss for the classification
- Experiments on 11 different organs with IoU > 58% (Pancreas, Bladder)



Journals & Magazines > IEEE Transactions on Medical ... > Volume: 38 Issue: 8

Efficient Multiple Organ Localization in CT Image Using 3D Region Proposal Network

Datasets Available

Publisher: IEEE

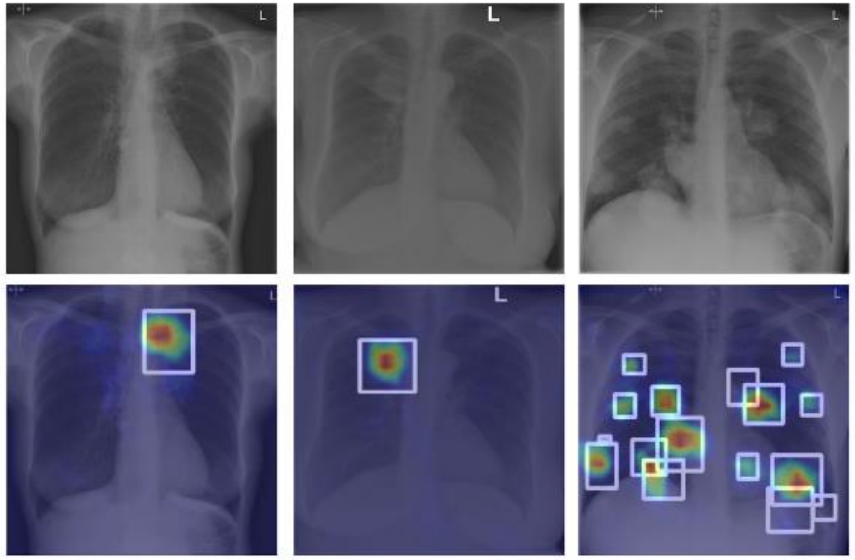
5 Author(s) Xuanang Xu ; Fugen Zhou ; Bo Liu ; Dongshan Fu ; Xiangzhi Bai **View All Authors**

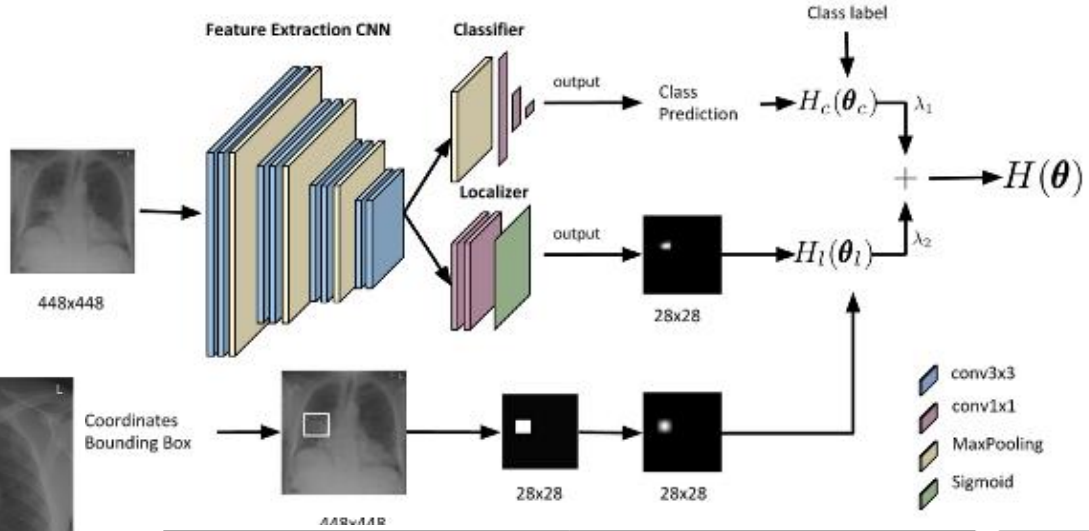
1 Paper Citation

1554 Full Text Views

Detection with Attention

- Detection of lesions
- Use a gaussian kernel to translate the bboxes to heatmaps and use them to calculate the saliency maps for each lesion.





Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Learning to detect chest radiographs containing pulmonary lesions using visual attention networks

Emanuele Pesce[a,1], Samuel Joseph Withey[b,d], Petros-Pavlos Ypsilantis[a], Robert Bakewell[c], Vicky Goh[b,d], Giovanni Montana[a,1,*]

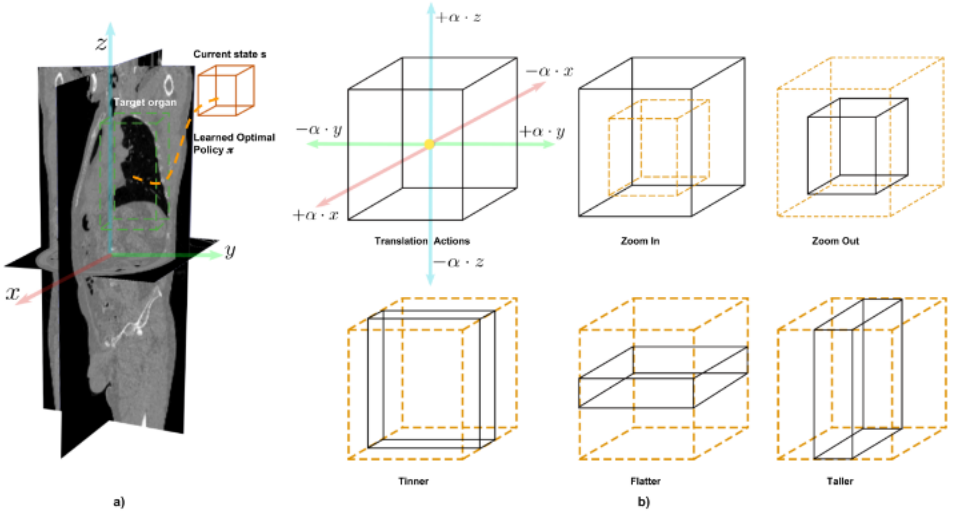
[a] Department of Biomedical Engineering, King's College London, London, UK
[b] Department of Radiology, Guy's & St Thomas' NHS Foundation Trust, London, UK
[c] Department of Medicine, Imperial College Healthcare NHS Trust, London, UK
[d] Department of Cancer Imaging, King's College London, London, UK

Detection with Reinforcement Learning

- The RL environment and action space is the 3D CT scan.
- The action space consists of 6 actions for translation, two for scaling the whole box and three to scale the box in each of the three directions.
- Based on Q-learning algorithm

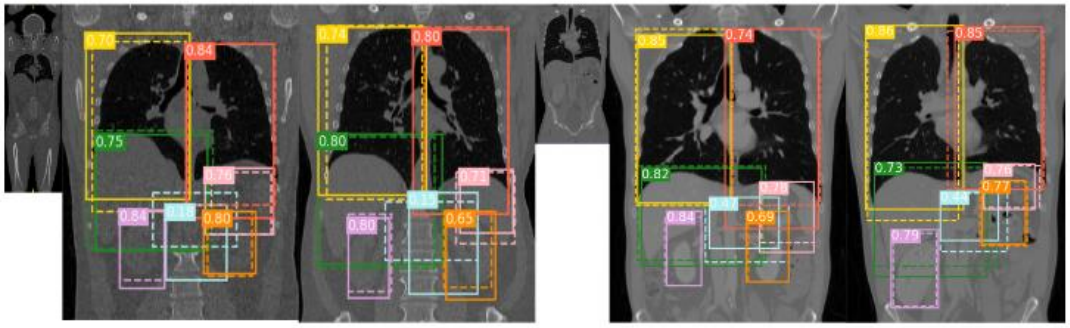
$$Q^*(s,a) = \max_{\pi} \mathbb{E}[R_t \mid s_t = s, a_t = a, \pi]$$

- R_t: expected reward, π policy



	Avg IoU	Wall dist [mm]	Centroid dist [mm]
Right Lung	0.77	3.46 ± 5.28	6.06 ± 10.25
Left Lung	0.73	4.91 ± 7.38	10.32 ± 17.09
Right Kidney	0.60	2.96 ± 2.91	5.69 ± 5.67
Left Kidney	0.57	4.06 ± 4.98	7.52 ± 9.02
Liver	0.80	2.41 ± 0.70	3.36 ± 1.34
Spleen	0.60	5.25 ± 7.23	9.20 ± 12.03
Pancreas	0.32	12.26 ± 13.60	20.79 ± 20.38
Global	0.63	5.04 ± 6.01	8.99 ± 10.82
Median	0.60	2.25	3.65

Proceedings of Machine Learning Research Accepted in MIDL 2020:1–11, 2020 Full Paper – MIDL 2020



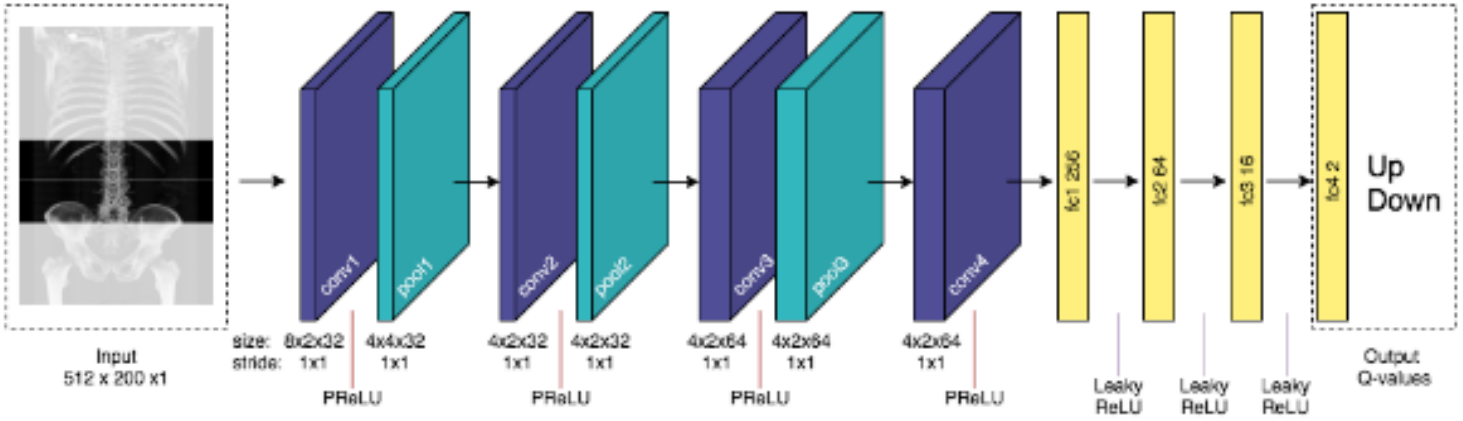
Deep Reinforcement Learning for Organ Localization in CT

Fernando Navarro[1] FERNANDO.NAVARRO@TUM.DE
Anjany Sekuboyina[1] ANJANY.SEKUBOYINA@TUM.DE
Diana Waldmannstetter[1] DIANA.WALDMANNSTETTER@TUM.DE
Jan C. Peeken[2] JAN.PEEKEN@TUM.DE
Stephanie E. Combs[2] STEPHANIE.COMBS@TUM.DE
Bjoern H. Menze[1] BJOERN.MENZE@TUM.DE
[1] *Department of Informatics And Mathematics, Technical University of Munich, Germany*
[2] *Department of Radio Oncology and Radiation Therapy, Klinikum rechts der Isar, Germany*

Editors: Accepted in MIDL 2020

Detection with Reinforcement Learning

- The RL for L3 Slice Localization in CT scans.
- A Deep Q-Network is trained to find the best policy to follow for this problem



Method	# of samples	Mean	Std	Median	Max	Error $> 10mm$
Interobserver	-	2.04	4.36	1.30	43.19	1
SC-Net [22]	-	6.78	13.96	**1.77**	46.98	12
L3UNet-2D [13]	900	4.24	6.97	2.19	40	**7**
Ours (Duel-DQN)	900	4.30	5.59	3	38	8
Ours	900	**3.77**	**4.71**	2.0	**24**	9
L3UNet-2D [13]	100	145.37	161.91	32.8	493	68
Ours	100	**5.65**	**5.83**	**4**	**26**	**19**
L3UNet-2D [13]	50	108.7	97.33	87.35	392.02	86
Ours	50	**6.88**	**5.79**	**6.5**	**26**	**11**
L3UNet-2D [13]	10	242.85	73.07	240.5	462	99
Ours	10	**8.97**	**8.72**	**7**	**56**	**33**

International Workshop on Machine Learning in Medical Imaging
MLMI 2021: Machine Learning in Medical Imaging pp 317-326 | Cite as

Deep Reinforcement Learning for L3 Slice Localization in Sarcopenia Assessment

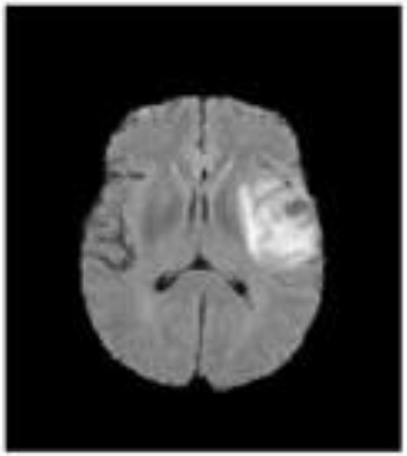
Authors Authors and affiliations
Othmane Laousy, Guillaume Chassagnon, Edouard Oyallon, Nikos Paragios, Marie-Pierre Revel, Maria Vakalopoulou

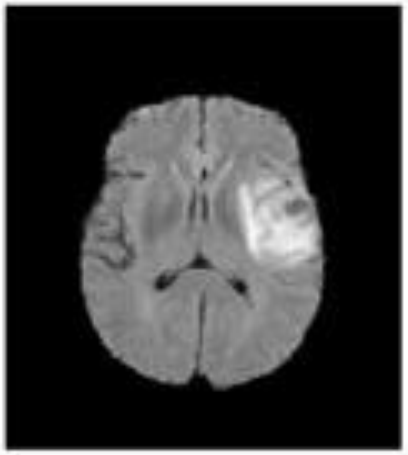
Conclusions Object Detection

- Object Detection is a very interesting topic for medical imaging
  - Annotations are very expensive and time consuming
  - Can provide a "weak" label for Deep Architectures
- Vision algorithms are usually used as baselines
- Variety of challenges for the medical imaging but the community is working on them!
- Detection of diseases is usually more challenging than detection of organs
  - Smaller size
  - Less data
  - Variance in appearance, size, ...

Deep Learning-based image denoising

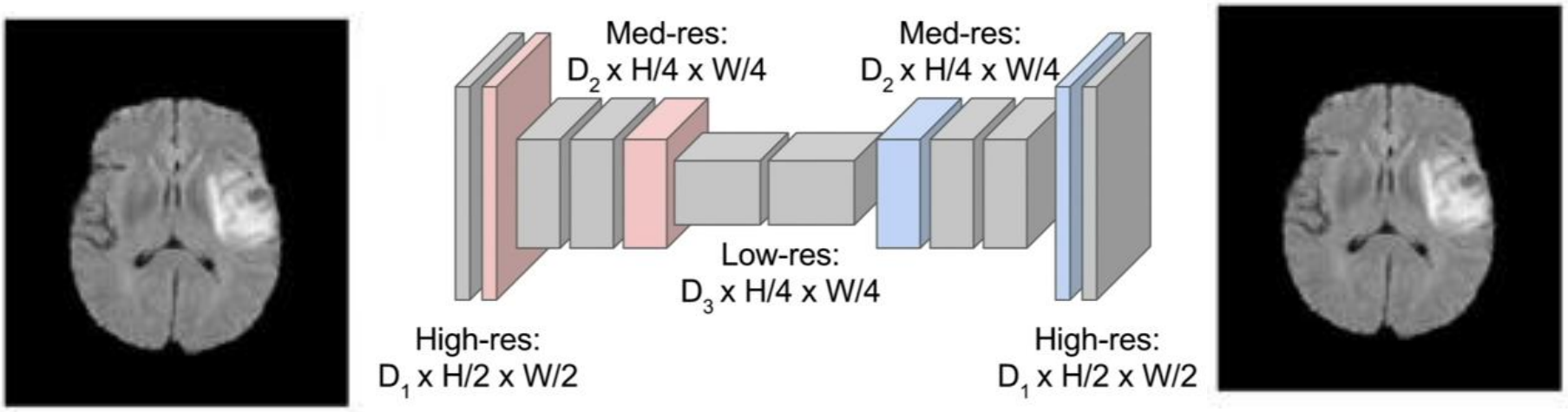
- Idea #2: Try to address the problem of denoising/ reconstruction as a semantic segmentation problem





CNN-based solution for Image Denoising

- Idea #1: Try to address the problem of denoising/ reconstruction as a semantic segmentation problem
  - Make use of the FCN architectures and autoencoders



- Use Downsampling and upsampling inside the network!

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Evaluation metrics

- For the quantitave evaluation of the results some of the metrics used are:
  - Mean Square Error (MSE)

$$MSE = \frac{1}{m\,n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

Evaluation metrics

- For the quantitave evaluation of the results some of the metrics used are:
  - Mean Square Error (MSE)

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

  - Peak Signal to Noise Ration (PSNR)

$$\begin{aligned} PSNR &= 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned}$$

MAX_I: Maximum possible pixel value of the image.

  - Structural Similarity Index (SSI)

$$SSIM(x,y) = l(x,y).\,c(x,y).\,s(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_x\sigma_y + c_2)(cov_{xy} + c_3)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)(\sigma_x\sigma_y + c_3)}$$

$Conv_{xy}$: covariance of x and y; L the dynamic range of values; $k_1 = 0.01$, $k_2 = 0.03$

$$c_1 = (k_1 L)^2,\ c_2 = (k_2 L)^2 \text{ et } c_3 = \frac{c_2}{2}$$

Evaluation metrics

- For the quantitave evaluation of the results some of the metrics used are:
  - Mean Square Error (MSE)

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

  - Peak Signal to Noise Ration (PSNR)

$$\begin{aligned} PSNR &= 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned}$$

MAX_I: Maximum possible pixel value of the image.

  - Structural Similarity Index (SSI)
  - Signal to Noise Ration (SNR)
  - Edge Preservation Index (EPI)
  - Coefficient of Correlation (CoC)

Some Losses for Image Denoising

- [Zhao et al. 2017]
  - The l_1 error

$$\mathcal{L}^{\ell_1}(P) = \frac{1}{N}\sum_{p\in P}|x(p)-y(p)|\,, \qquad \partial\mathcal{L}^{\ell_1}(P)/\partial x(p) = \mathrm{sign}\,(x(p)-y(p))\,.$$

  - The l_2 error

Some Losses for Image Denoising

- [Zhao et al. 2017]
  - The l_1 error

$$\mathcal{L}^{\ell_1}(P) = \frac{1}{N}\sum_{p\in P}|x(p)-y(p)|, \qquad \partial\mathcal{L}^{\ell_1}(P)/\partial x(p) = \text{sign}\,(x(p)-y(p)).$$

  - The l_2 error
  - The SSIM

$$\begin{aligned}\text{SSIM}(p) &= \frac{2\mu_x\mu_y + C_1}{\mu_x^2+\mu_y^2+C_1}\cdot\frac{2\sigma_{xy}+C_2}{\sigma_x^2+\sigma_y^2+C_2} \\ &= l(p)\cdot cs(p)\end{aligned} \qquad \mathcal{L}^{\text{SSIM}}(P) = \frac{1}{N}\sum_{p\in P}1-\text{SSIM}(p).$$

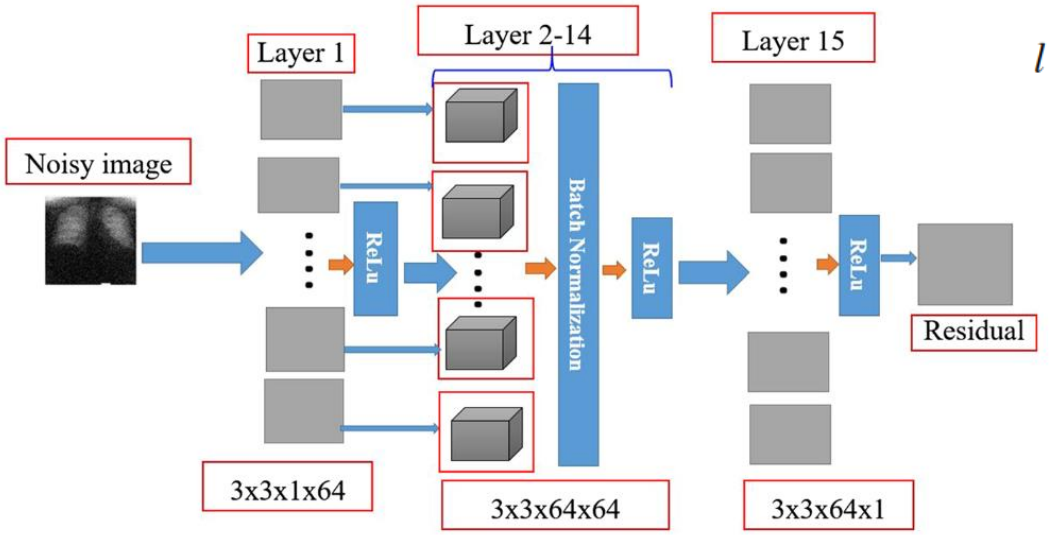
$$\begin{aligned}&\frac{\partial\mathcal{L}^{\text{MS-SSIM}}(P)}{\partial x(q)} \\ &= \left(\frac{\partial l_M(\tilde{p})}{\partial x(q)} + l_M(\tilde{p})\cdot\sum_{i=0}^{M}\frac{1}{cs_i(\tilde{p})}\frac{\partial cs_i(\tilde{p})}{\partial x(q)}\right)\cdot\prod_{j=1}^{M}cs_j(\tilde{p}),\end{aligned}$$

  - Advesarial Losses

More Papers

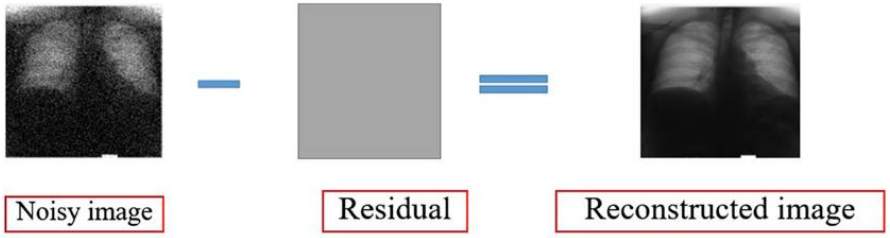
Denoising on XRays

- [Sori et al. 2017] Design of a deep feed forward CNN model which directly approximate the noise from a noisy image. Residual learning is approximated, and batch normalization is also incorporated to boost model performance.



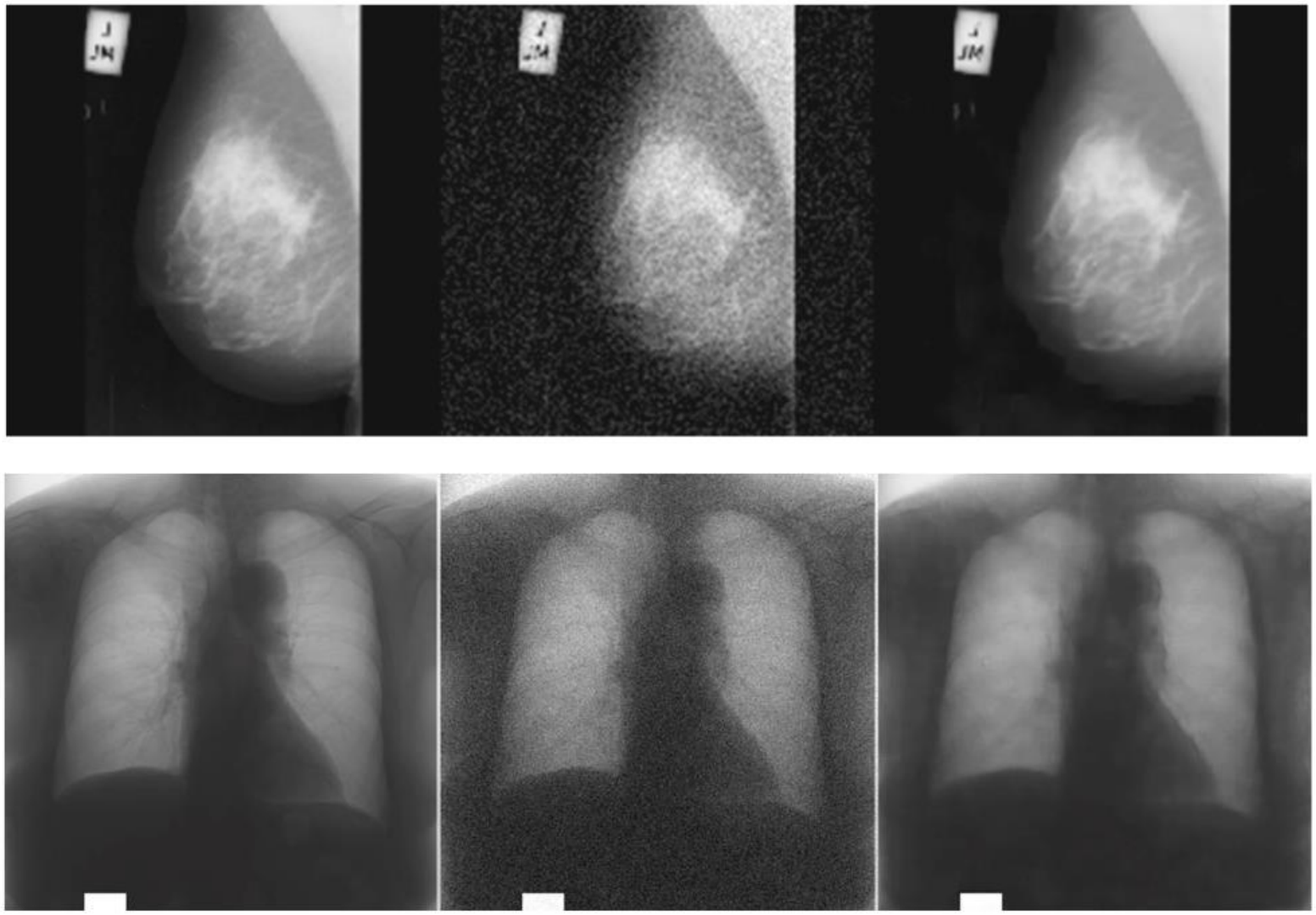
$$l(\lambda) = \frac{1}{2N} \sum_{k=1}^{N} \| \Re(z_k; \lambda) - (z_k - x_k) \|_F^2$$

Fig. 1 Residual learning phase of the model for medical image denoising



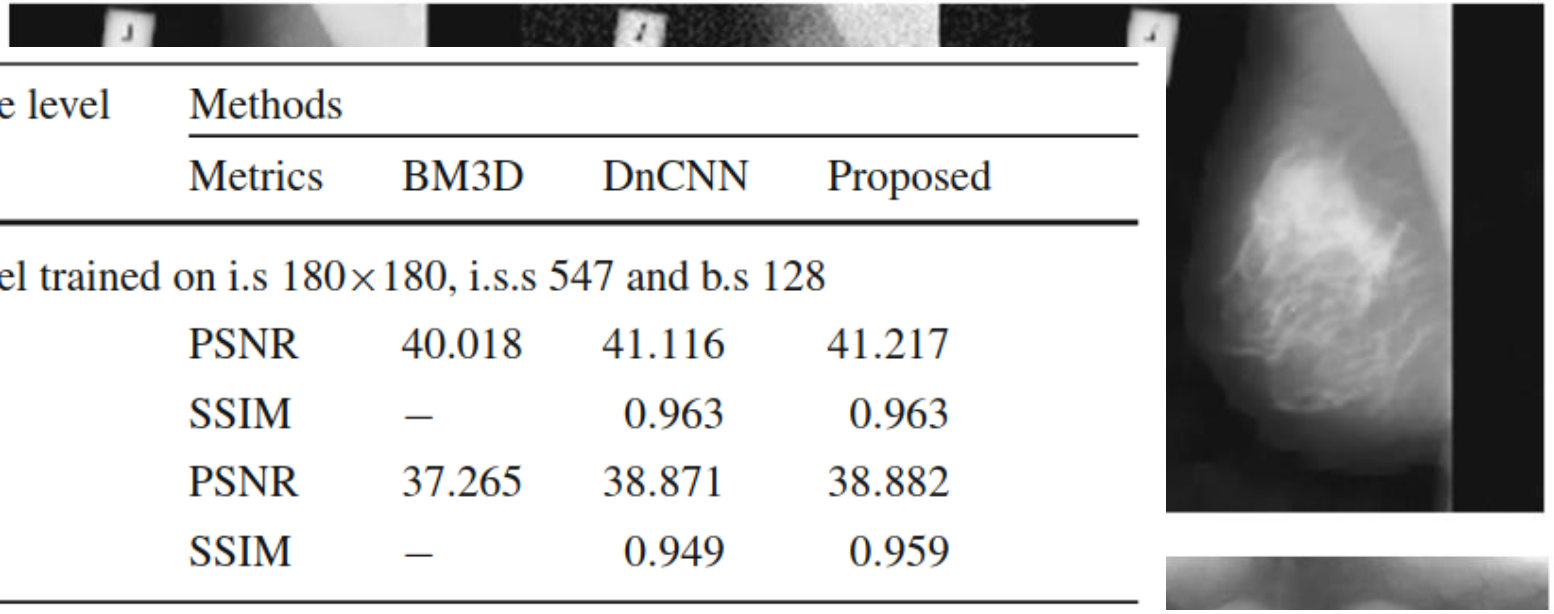
Denoising on XRays

- [Sori et al. 2017]

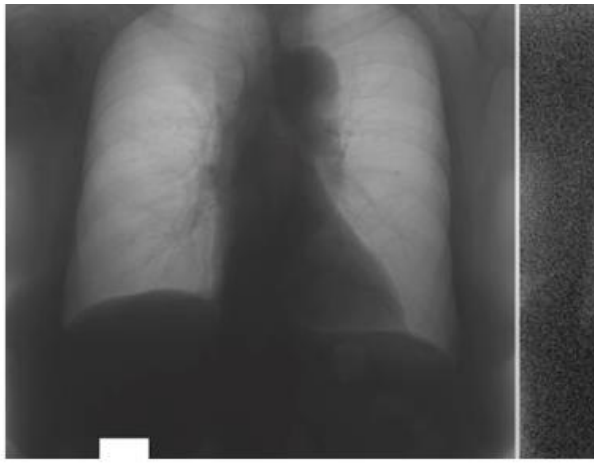


Denoising on XRays

- [Sori et al. 2017]



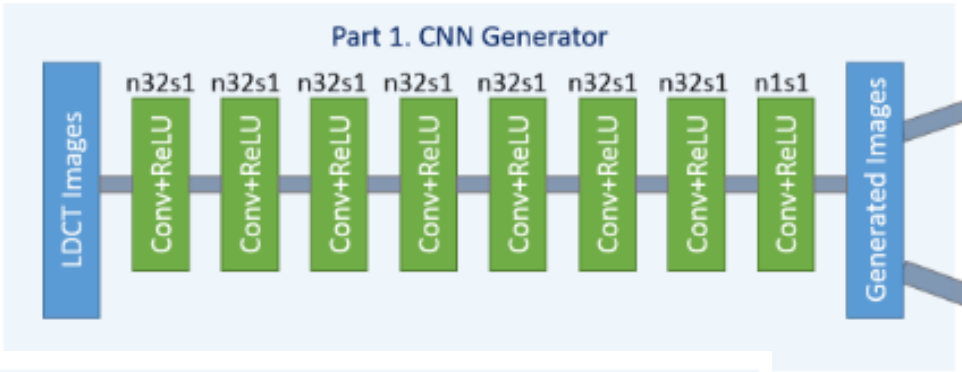
Noise level	Methods			
	Metrics	BM3D	DnCNN	Proposed
Model trained on i.s 180×180, i.s.s 547 and b.s 128				
15	PSNR	40.018	41.116	41.217
	SSIM	–	0.963	0.963
25	PSNR	37.265	38.871	38.882
	SSIM	–	0.949	0.959

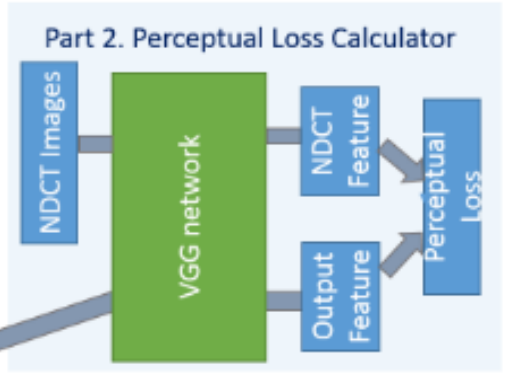


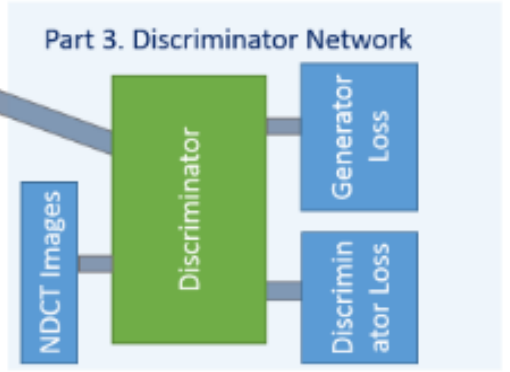
Noise level	Methods			
	Metrics	CNN DAE	DnCNN	Proposed
Model trained on i.s 64×64, i.s.s 235 and b.s 10				
15	PSNR	–	39.246	39.250
	SSIM	0.89	0.950	0.950
25	PSNR	–	36.696	36.70
	SSIM	0.89	0.932	0.932

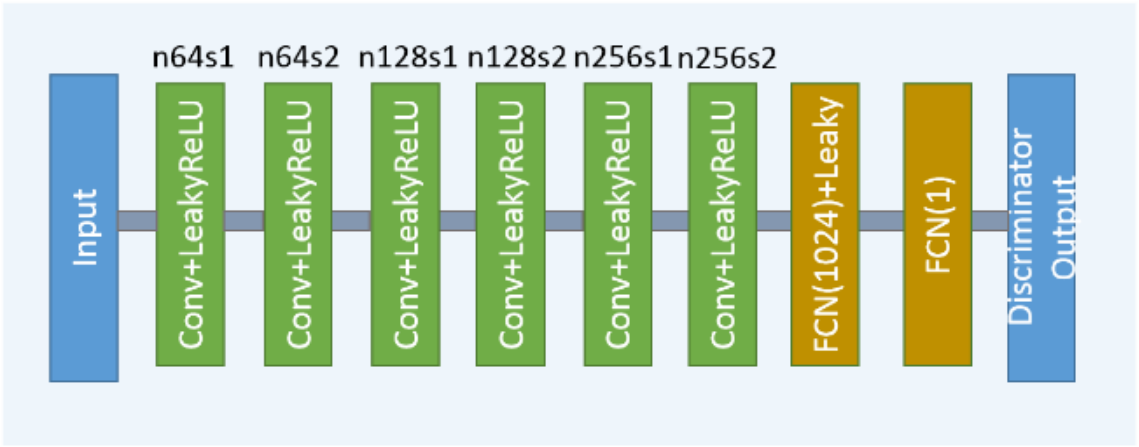
Low Dose CT Image Denoising

- [Yang et al. 2018] Computed tomography (CT) is one of the most important modalities. Potential radiation risk to the patient since x-rays could cause generic damage
- Lower radiation increases noise and artifacts









Low Dose CT Image Denoising

- [Yang et al. 2018]

	Fig. 5		Fig. 7	
	PSNR	SSIM	PSNR	SSIM
LDCT	19.7904	0.7496	18.4519	0.6471
CNN-MSE	24.4894	0.7966	23.2649	0.7022
WGAN-MSE	24.0637	0.8090	22.7255	0.7122
CNN-VGG	23.2322	0.7926	22.0950	0.6972
WGAN-VGG	23.3942	0.7923	22.1620	0.6949
WGAN	22.0168	0.7745	20.9051	0.6759
‘1 GAN	21.8676	0.7581	21.0042	0.6632
DictRecon	24.2516	0.8148	24.0992	0.7631

	Fig. 5		Fig. 7	
	Mean	SD	Mean	SD
NDCT	9	36	118	38
LDCT	11	74	118	66
CNN-MSE	12	18	120	15
WGAN-MSE	9	28	115	25
CNN-VGG	4	30	104	28
WGAN-VGG	9	31	111	29
WGAN	23	37	135	33
GAN	8	35	110	32
DictRecon	4	11	111	13

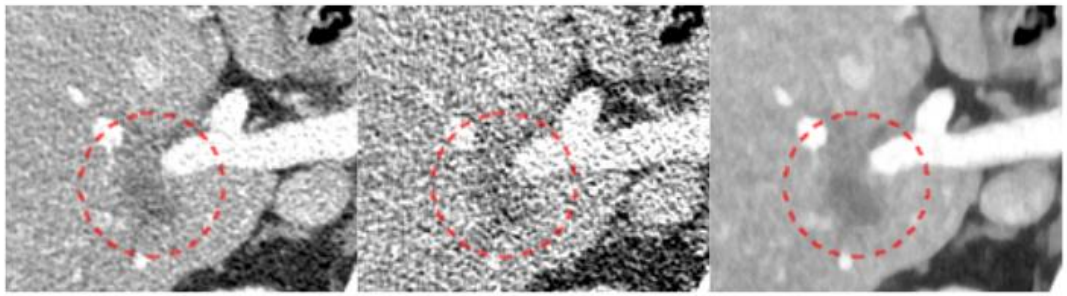
(a) Full Dose FBP (b) Quarter Dose FBP (c) DictRecon

(d) GAN (e) CNN-MSE (f) CNN-VGG

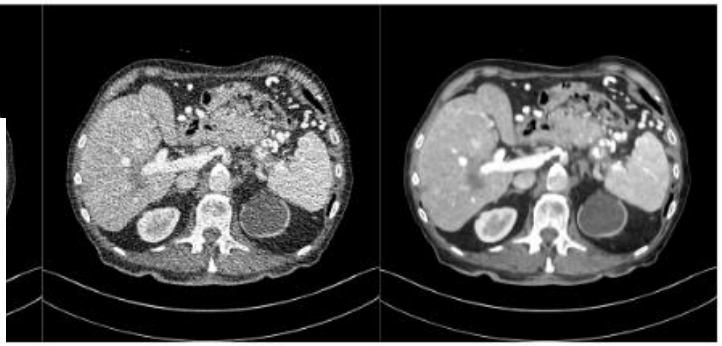
(g) WGAN (h) WGAN-MSE (i) WGAN-VGG

Low Dose CT Image Denoising

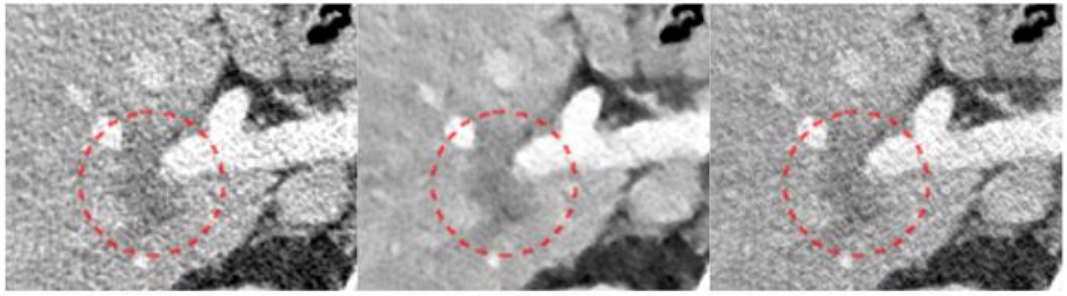
- [Yang et al. 2018]



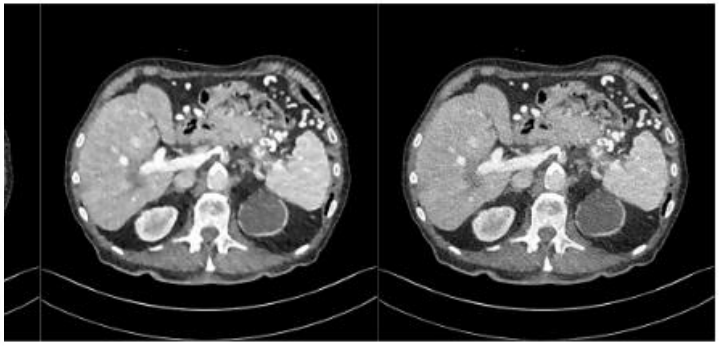
(a) Full Dose FBP (b) Quarter Dose FBP (c) DictRecon



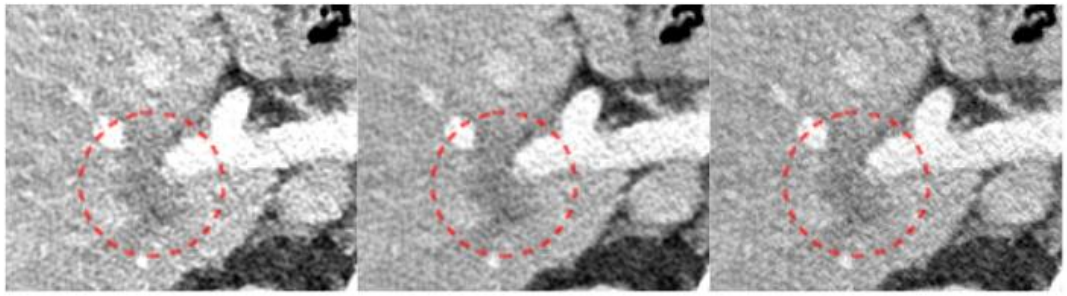
(b) Quarter Dose FBP (c) DictRecon



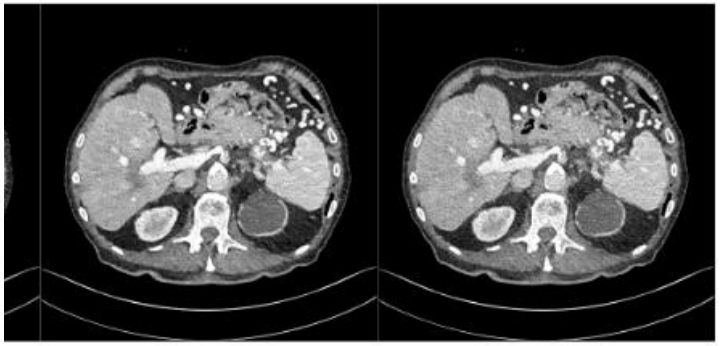
(d) GAN (e) CNN-MSE (f) CNN-VGG



(e) CNN-MSE (f) CNN-VGG



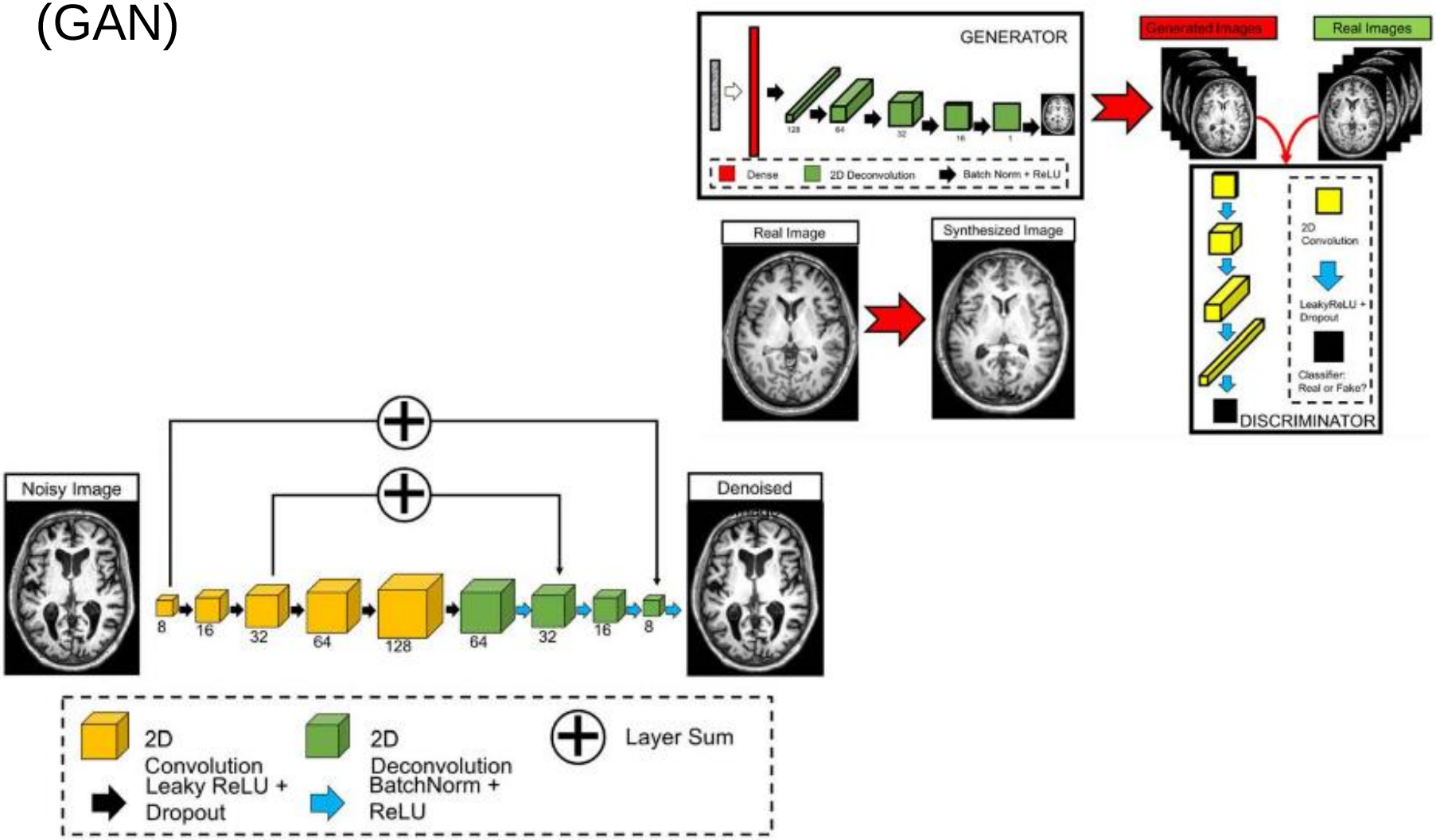
(g) WGAN (h) WGAN-MSE (i) WGAN-VGG



(h) WGAN-MSE (i) WGAN-VGG

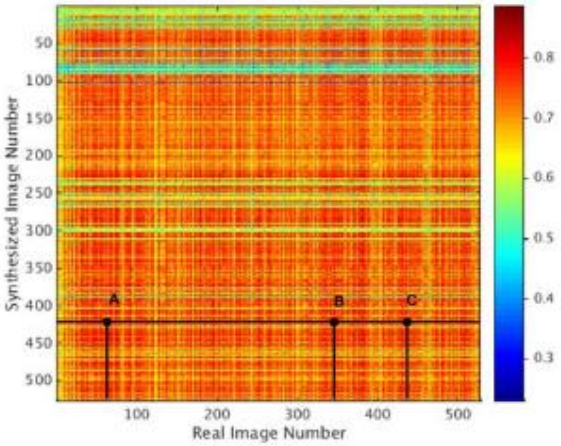
Learning Brain MRI Manifolds

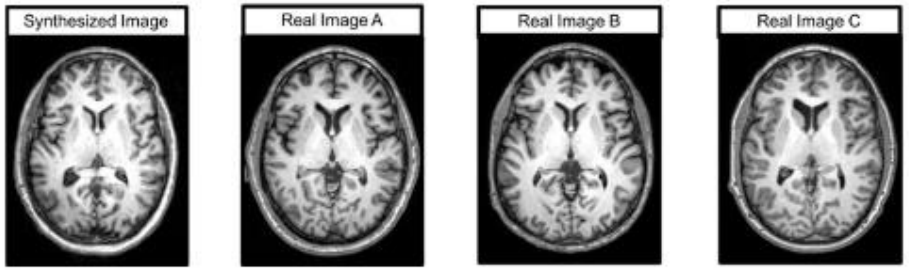
- [Bernudez et al. 2018] Unsupervised synthesis of T1-weighted brain MRI using a Generative Adversarial Network (GAN)

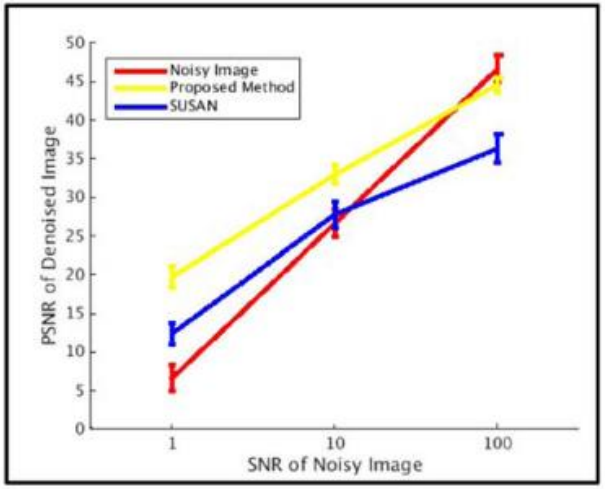


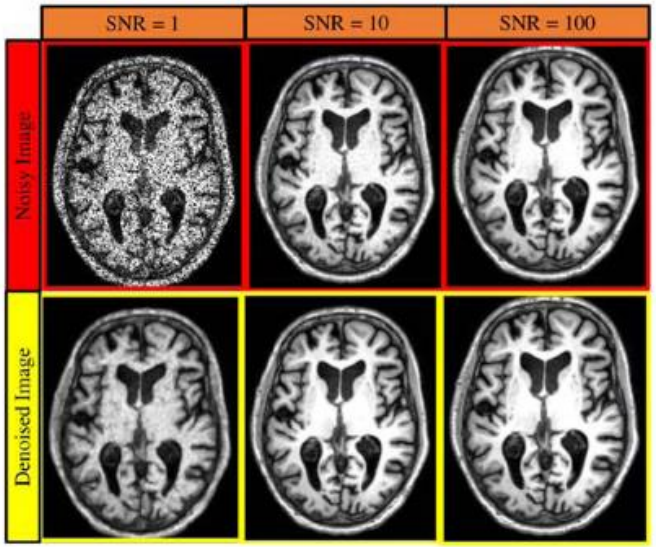
Learning Brain MRI Manifolds

- [Bernudez et al. 2018] Unsupervised synthesis of T1-weighted brain MRI using a Generative Adversarial Network (GAN)









A CNN to filter artifacts

- [Gurbani et al. 2017] A deep learning model was developed that was capable of identifying and filtering out poor quality spectra. The core of the model used a tiled CNN that analyzed frequency-domain spectra to detect artifacts.
- The Cho/NAA ratio is widely used for depiction of tumor volumes and infiltration as a result of increased contrast caused by the opposite changes of these metabolites in the tumor.

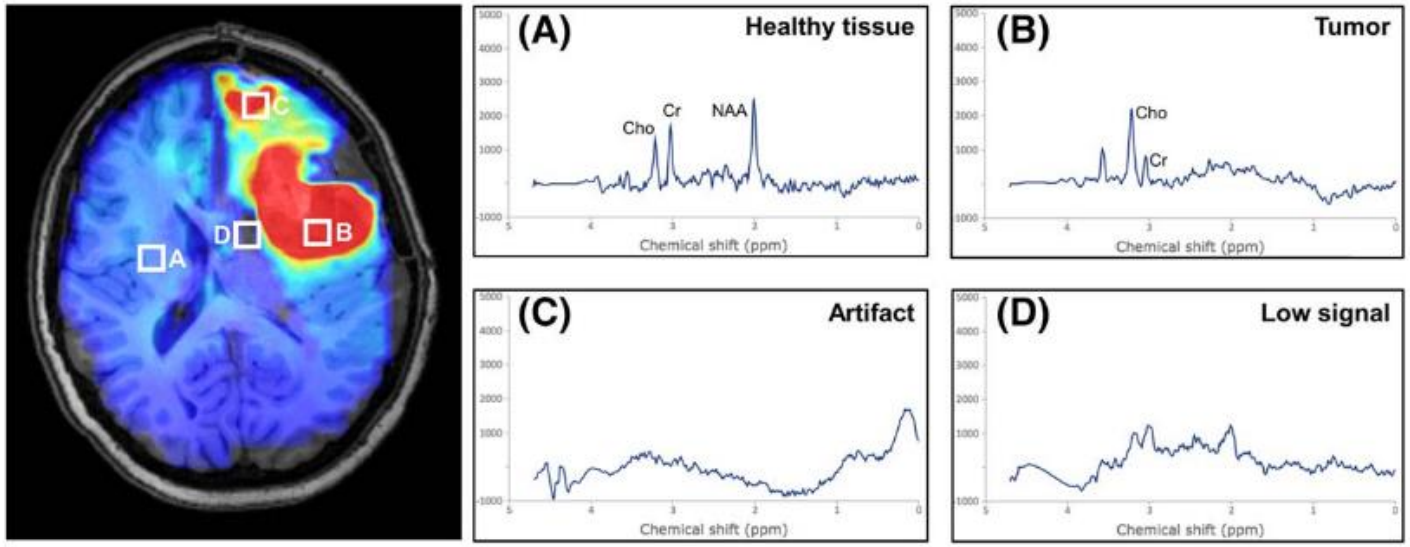


FIGURE 1 Artifacts in MRSI arise for several reasons and can lead to false interpretation of pathology. A, Healthy tissue shows a relatively low Cho/NAA ratio. B, Tumor shows an elevated ratio, appearing as hyperintense on a Cho/NAA map. Artifacts can arise in tissue boundaries and in areas with poor lipid or water suppression, and can result in either hyperintense lesions (C) or dropout of signal (D)

A CNN to filter artifacts

- [Gurbani et al. 2017] A deep learning model was developed that was capable of identifying and filtering out poor quality spectra. The core of the model used a tiled CNN that analyzed frequency-domain spectra to detect artifacts.

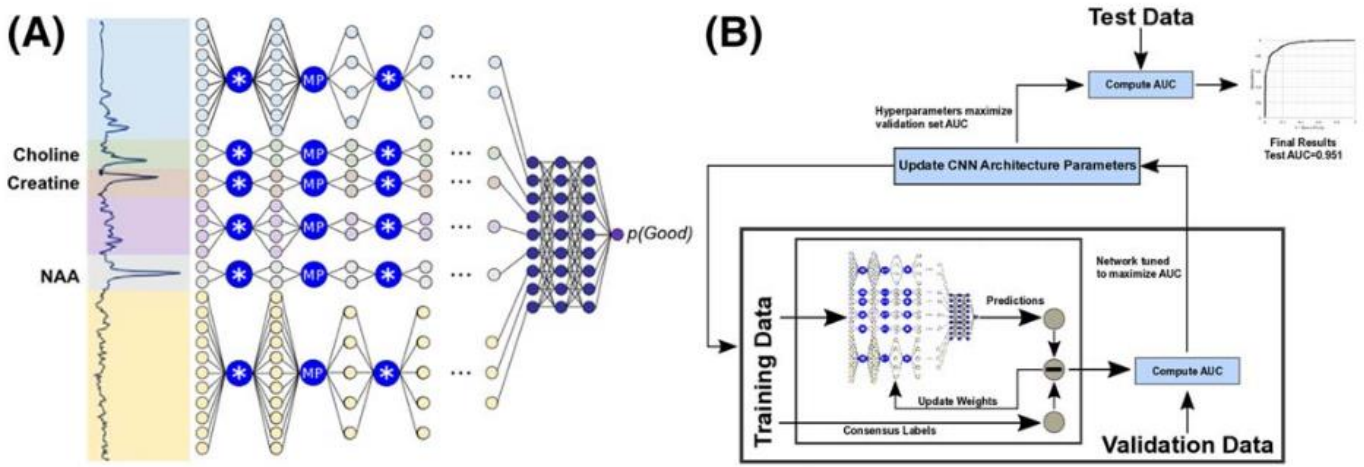
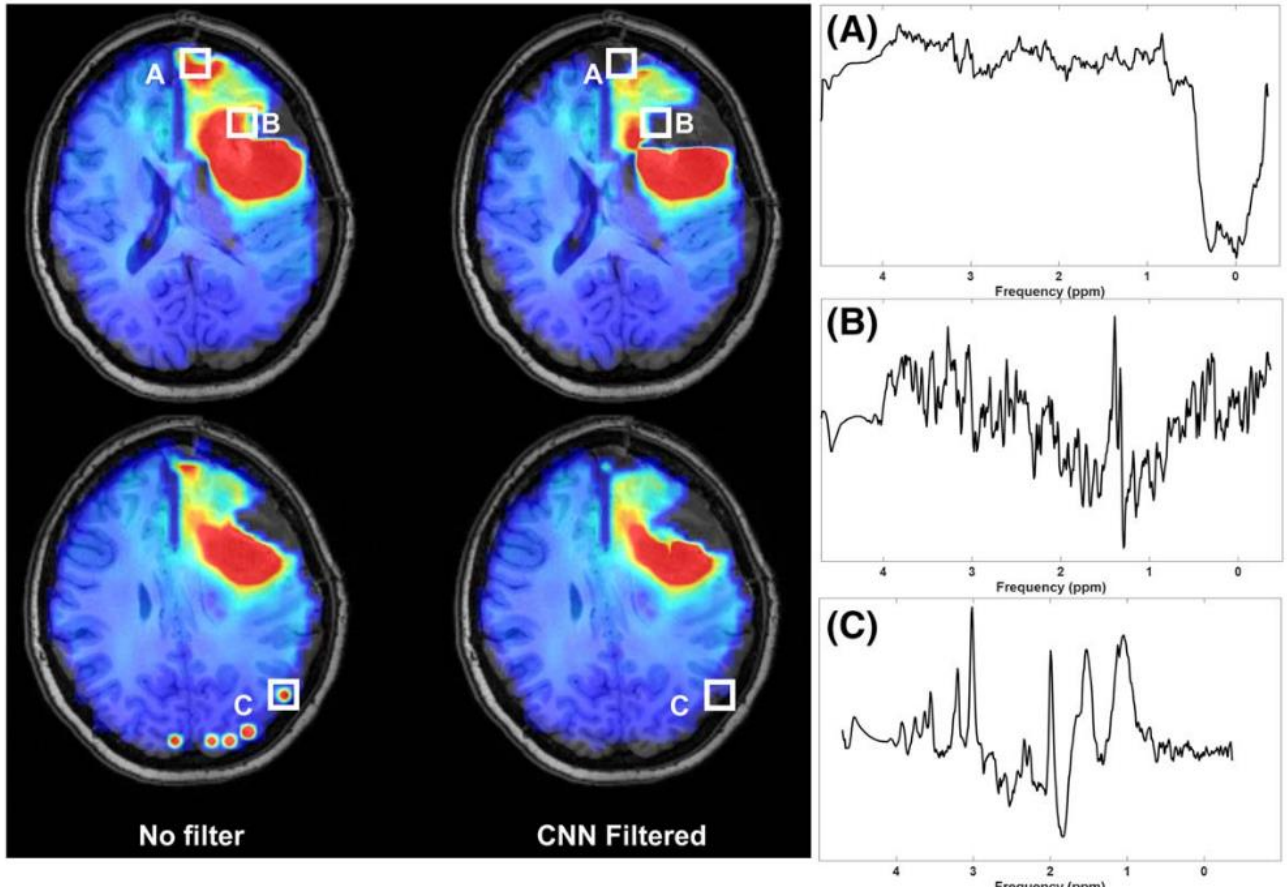


FIGURE 3 A, High-level overview of the convolutional neural network (CNN) for spectral quality analysis. Input spectra are split into 6 tiles and passed through a series of convolution (*) and max-pooling (MP) layers, then concatenated and passed through fully connected layers to generate a scalar output of spectral quality. B, Bayesian optimization is used to iteratively optimize architecture hyperparameters. AUC, area under the curve

A CNN to filter artifacts

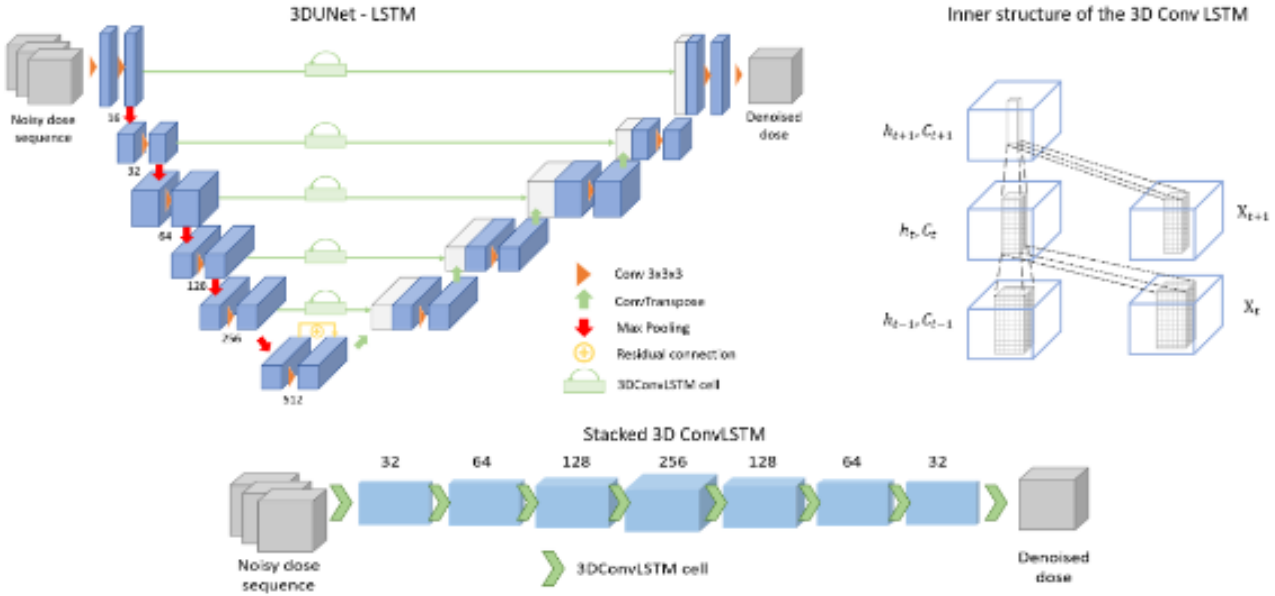
- [Gurbani et al. 2017] A deep learning model was developed that was capable of identifying and filtering out poor quality spectra. The core of the model used a tiled CNN that analyzed frequency-domain spectra to detect artifacts.



Dose denoising

- [Martinot et al. 2021] Proposes a ConvLSTM to handle 3D data and introduce a 3D recurrent and fully convolutional neural network architecture.

$$\mathcal{L} = \sum_{i=0}^{N_{samples}} \left(\left\| X_{N_{T+1}}^{(i,estimated)} - X_{N_{T+1}}^{i} \right\|_1 + SSIM\left(X_{N_{T+1}}^{(i,estimated)}, X_{N_{T+1}}^{i} \right) \right)$$



$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \odot C_{t-1} + b_i)$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + Wcf \odot C_{t-1} + b_f)$$

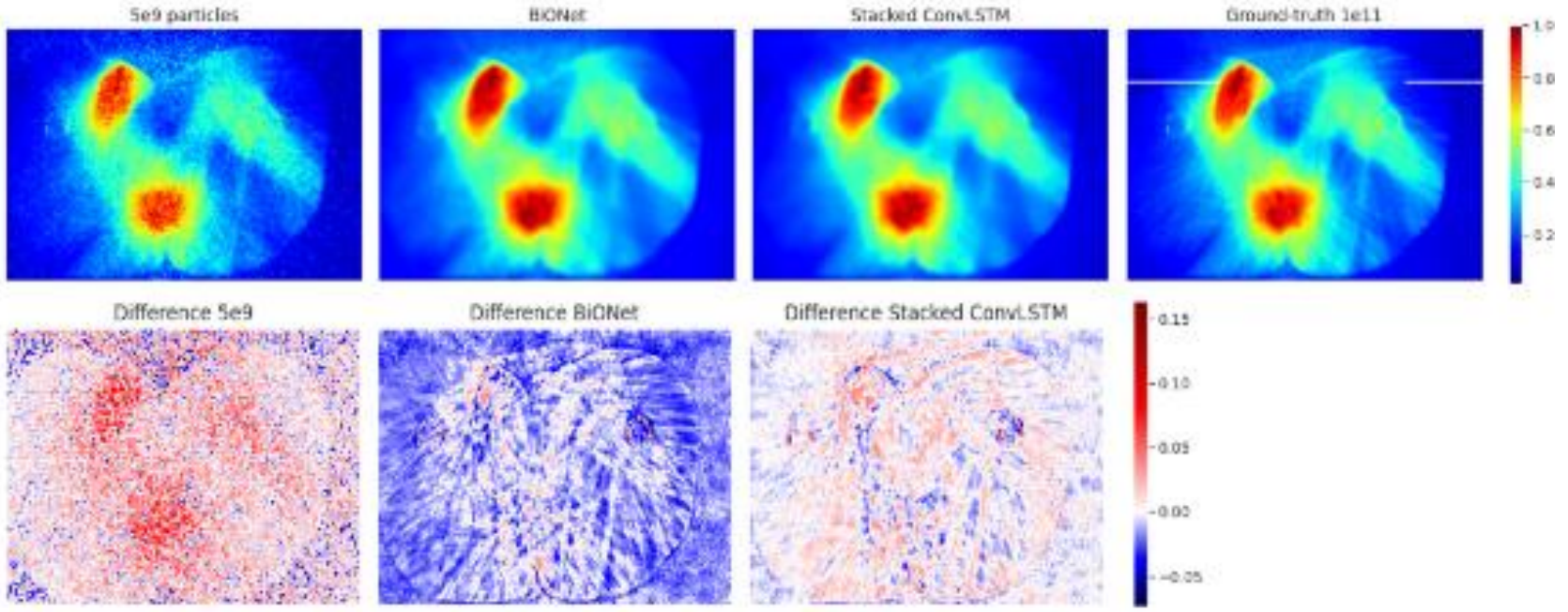
$$C_t = f_t \odot C_{t-1} + i_t \odot tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c)$$

$$o_t = \sigma(W_{xo} * X_t + W_{co} \odot C_t + b_o)$$

$$H_t = o_t \odot tanh(C_t)$$

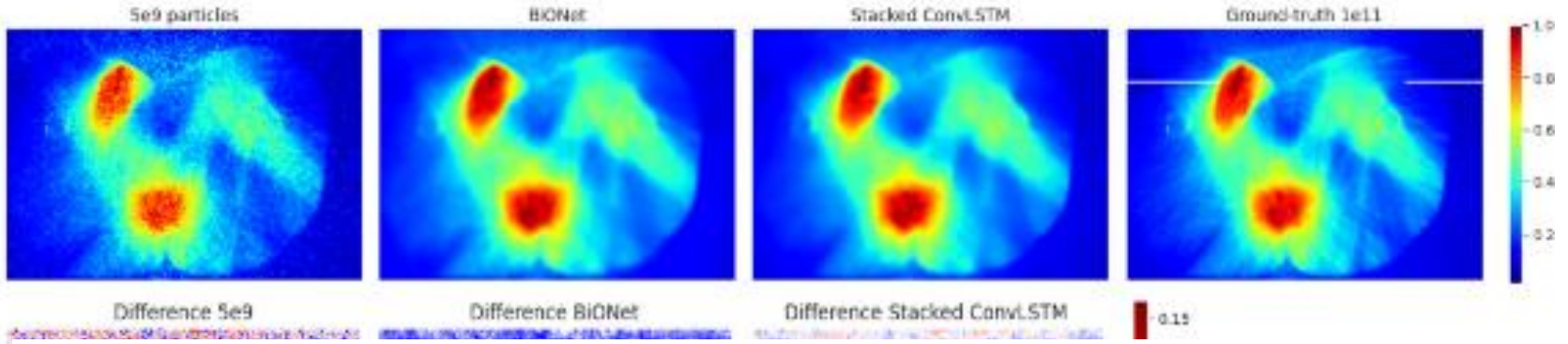
Dose denoising

- [Martinot et al. 2021] Proposes a ConvLSTM to handle 3D data and introduce a 3D recurrent and fully convolutional neural network architecture.



Dose denoising

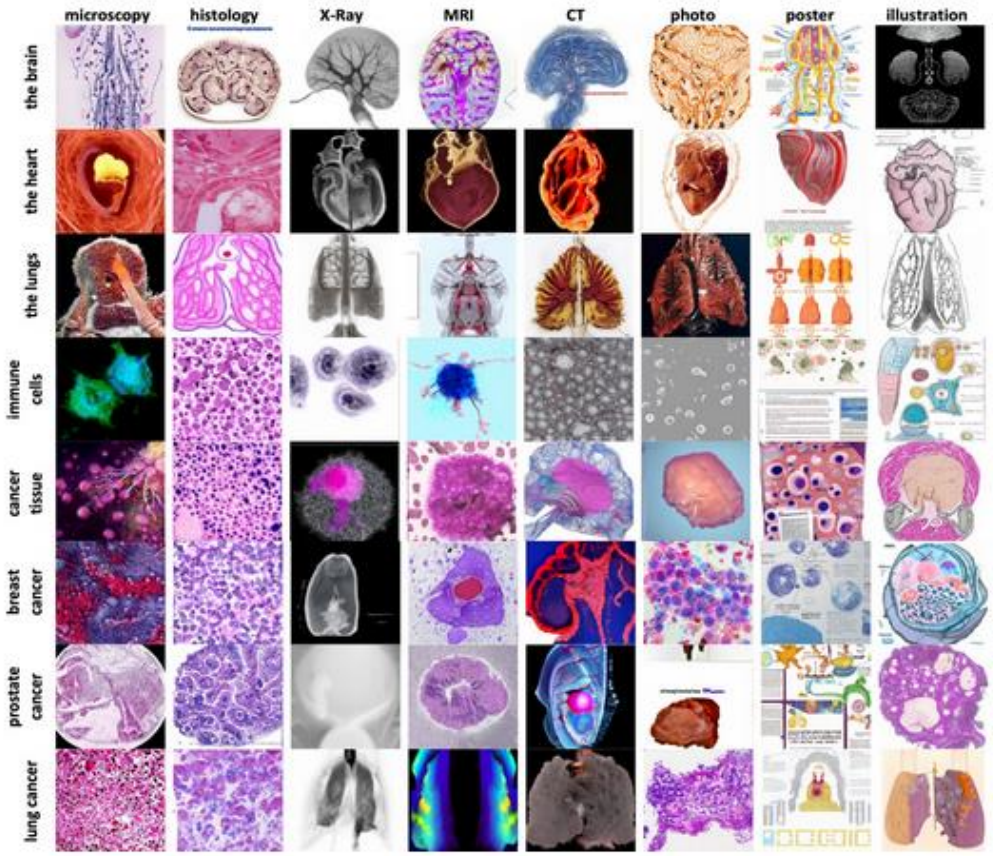
- [Martinot et al. 2021] Proposes a ConvLSTM to handle 3D data and introduce a 3D recurrent and fully convolutional neural network architecture.



Method	SSIM	GPR	L1	# parameters
Inputs 5e9 particles	58.1 ± 0.1	59.1 ± 2.1	0.149 ± 0.050	
3DUNet [14]	80.0 ± 2.4	61.2 ± 2.8	0.088 ± 0.007	10 M
Pix2Pix 3D [7]	55.4 ± 8.6	66.6 ± 14.4	0.102 ± 0.009	120 M
3D BiONet [19]	93.0 ± 0.2	90.6 ± 1.2	0.080 ± 0.001	178 M
Proposed 3DUNet ConvLSTM	64.5 ± 6.1	79.1 ± 1.2	0.037 ± 0.004	36 M
Proposed Stacked 2D ConvLSTM	81.6 ± 3.2	74.1 ± 3.1	0.021 ± 0.003	1.5 M
Proposed Stacked 3D ConvLSTM	**97.9 ± 0.9**	**94.1 ± 1.2**	**0.004 ± 0.001**	5 M

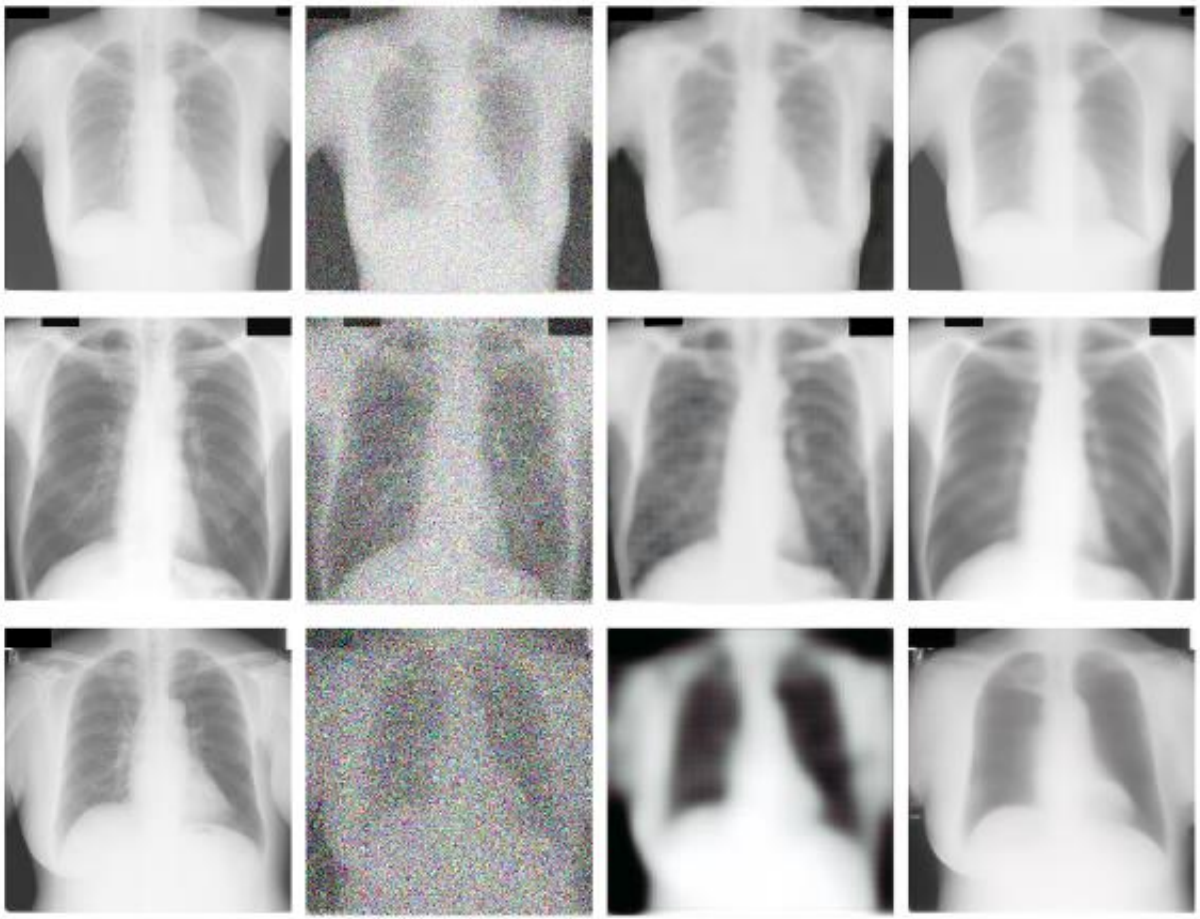
What about recent powerful models?

- Diffusion models are currently very popular generative models trained on reconstruction/ denoising.
- [Kather et al. 2022] GLIDE model: a diffusion model for the problem of text-conditional synthesis and compare two different guidance strategies: CLIP guidance and classifier-free guidance. [December 2021]
- Dataset: 250 million text-image from the internet incorporating Conceptual Captions, the text-image pairs from Wikipedia and a subset of YFCC100M.



What about recent powerful models?

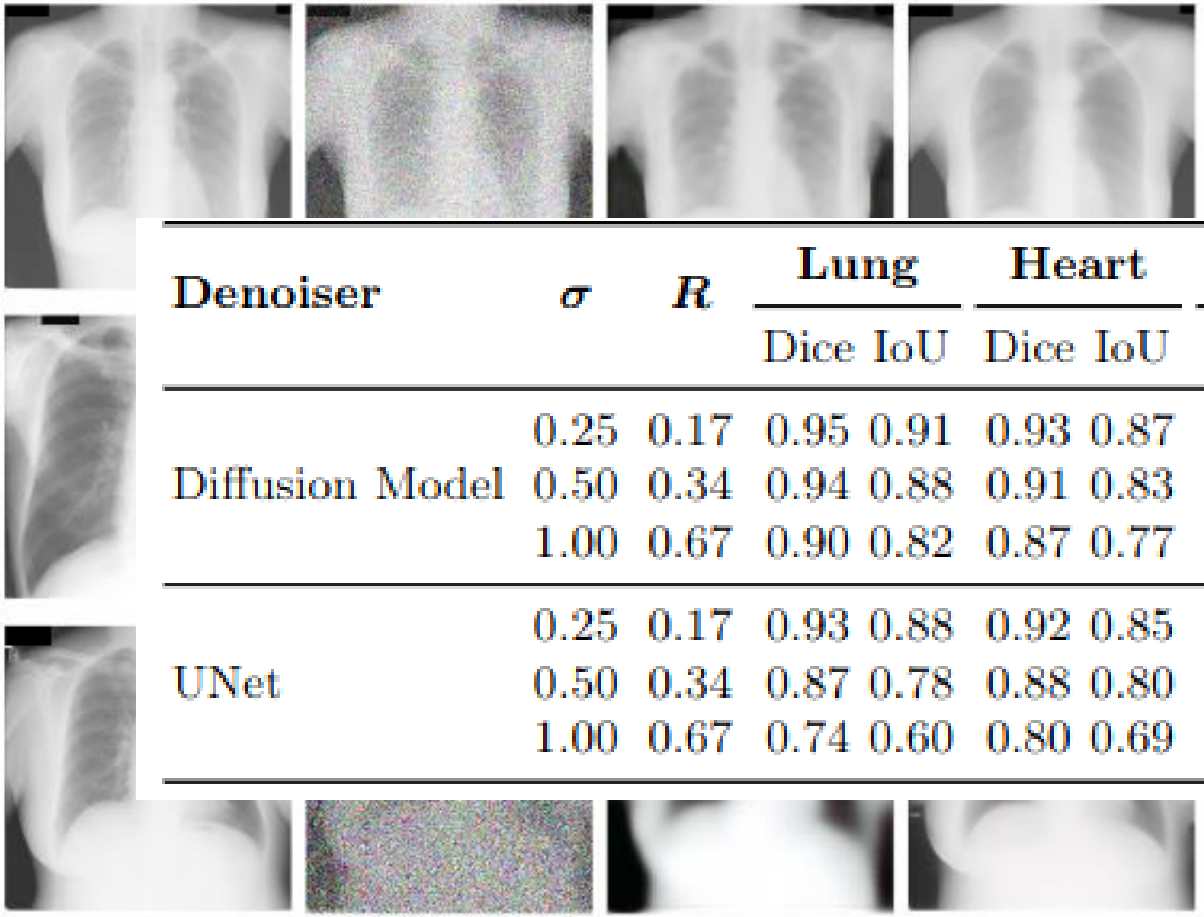
- [Laoussy et al. 2023] Use of the Denoising Diffusion Probabilistic models for Xrays.



Results for different levels of noise
σ = 0.25, 0.5, and 1.0.

What about recent powerful models?

- [Laoussy et al. 2023] Use of the Denoising Diffusion Probabilistic models for Xrays.



Results for different levels of noise

Denoiser	σ	R	Lung		Heart		Clavicles		
			Dice	IoU	Dice	IoU	Dice	IoU	%⊘
Diffusion Model	0.25	0.17	0.95	0.91	0.93	0.87	0.78	0.65	0.05
	0.50	0.34	0.94	0.88	0.91	0.83	0.63	0.48	0.08
	1.00	0.67	0.90	0.82	0.87	0.77	0.28	0.19	0.12
UNet	0.25	0.17	0.93	0.88	0.92	0.85	0.73	0.59	0.06
	0.50	0.34	0.87	0.78	0.88	0.80	0.54	0.41	0.11
	1.00	0.67	0.74	0.60	0.80	0.69	0.18	0.11	0.18

ıd 1.0.

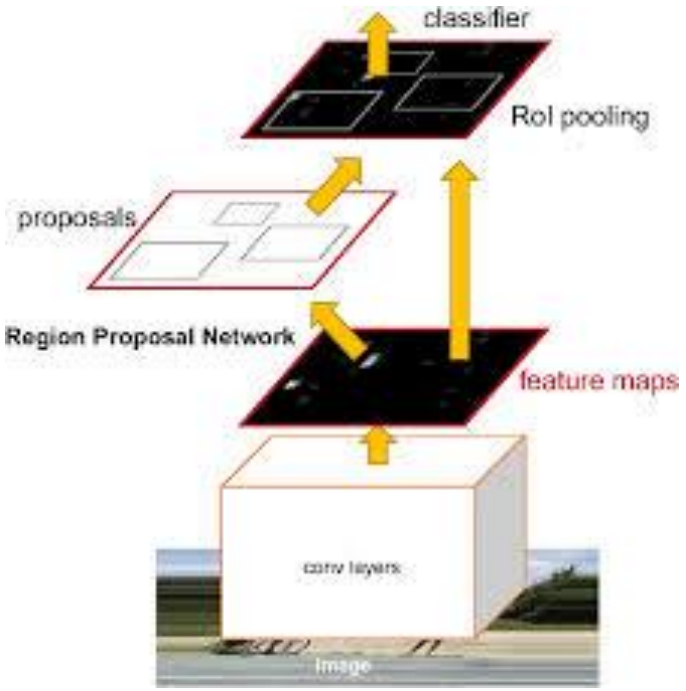
Input Image Noised Image UNet model Diffusion

Conclusions Image Denoising

- Image reconstruction and denoising is very important for medical imaging.
- Variety of metrics can be used for the evaluation of the denoised image.
- Very important to find formulations that can be verified from the physics or well established theories.
- Unsupervised methods and adversarial networks will play a significant role in the future. Really active research area.
- Challenges
  - Learning the right features
  - Detecting when it goes wrong
  - Going beyond human-level performance

Lab Session!

- Detecting lesions in mammograms
  - Faster R-CNN



jupyter

CO

PyTorch

TorchVision

Facebook Open Source

Deep learning for medical imaging

Olivier Colliot, PhD
Research Director at CNRS
Co-Head of the ARAMIS Lab – www.aramislab.fr
PRAIRIE – Paris Artificial Intelligence Research Institute

Maria Vakalopoulou, PhD
Assistant Professor at CentraleSupelec
Mathematics and Informatics (MICS)
Office: Bouygues Building Sb.132

Master 2 - MVA