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

Course website: http://www.aramislab.fr/teaching/DLMI-2019-2020/
Piazza (for registered students): https://piazza.com/centralesupelec/spring2020/mvadlmi/

Some Updates

- The project description is out!
  - On piazza resources you can find the description!
  - You should split on teams of 2-3 people

- Deadline for project proposal: **21 February**
  - Ask us for possible topics or in case you have difficulties to propose a subject/ find a team!

Acknowledgements

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

Thank you!!

Previous Lecture
Classification & Regression

Classification

Outputs y are classes: **classification** problem

Beavers

Chairs

Trees

Labelled training examples

Learning

Input

x

f

$y = f(x)$

Beaver

Output

y

$$\hat{f} = arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell\left(y^{(i)}, f(x^{(i)})\right)$$

Loss function:
measures error in prediction

Classification and Regression

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

Input features

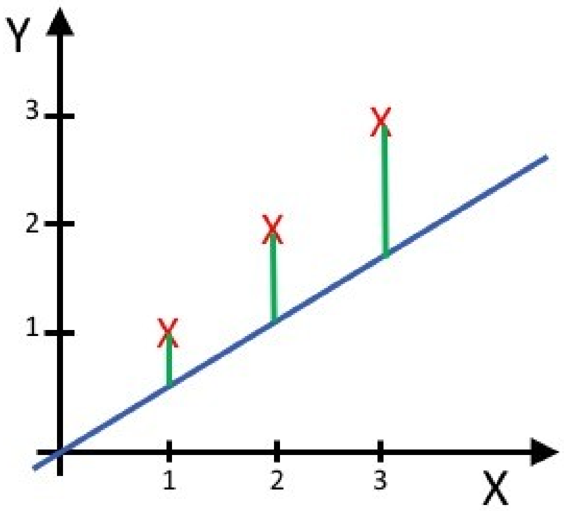
Output: y

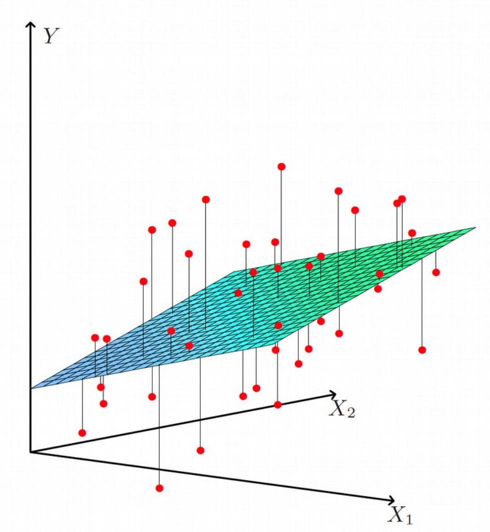
Model: f, $y = f(x)$

The "artificial intelligence"

Loss: $\ell(y, x)$

Quantifies how much the prediction is far from the true output





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

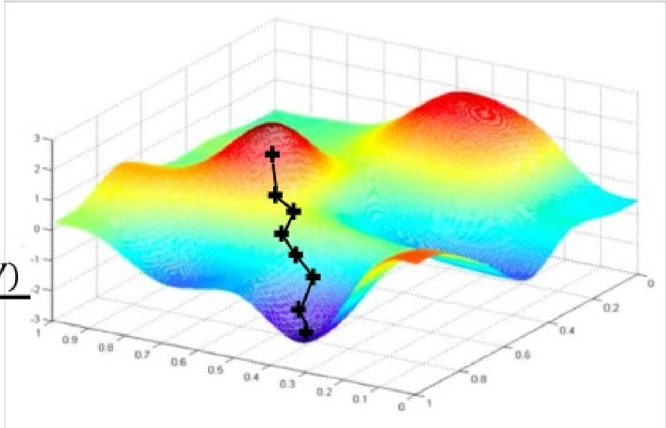
Optimization algorithm: method to find the minimum

Stochastic gradient descent

Stochastic gradient descent with several samples (mini-batch)

Algorithm

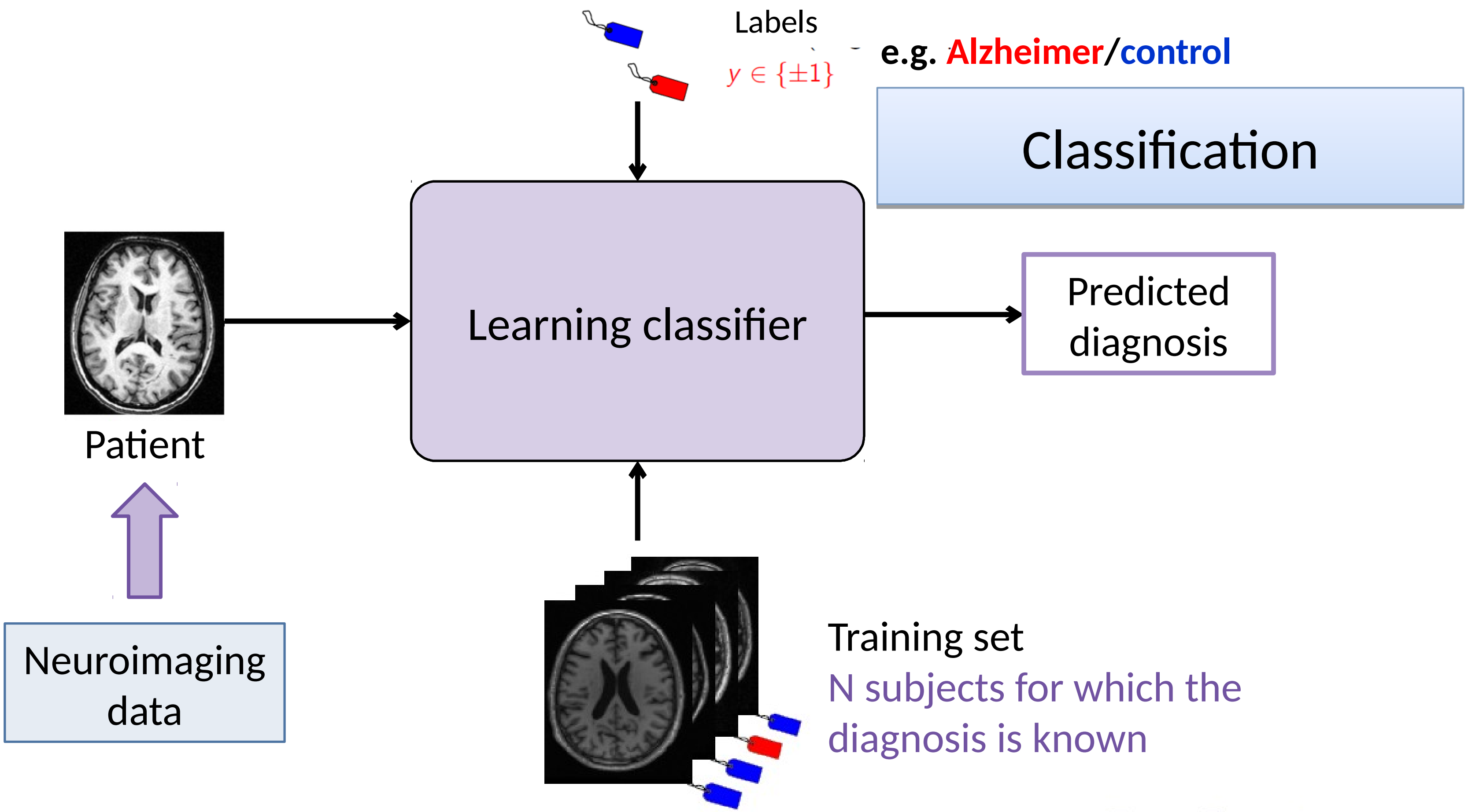
1. Initialize weights randomly $\sim\mathcal{N}(0, \sigma^2)$
2. Loop until convergence:
3. Pick batch of B data points
4. Compute gradient, $\frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}} = \frac{1}{B}\sum_{k=1}^{B} \frac{\partial J_k(\boldsymbol{W})}{\partial \boldsymbol{W}}$
5. Update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

Example in brain image classification



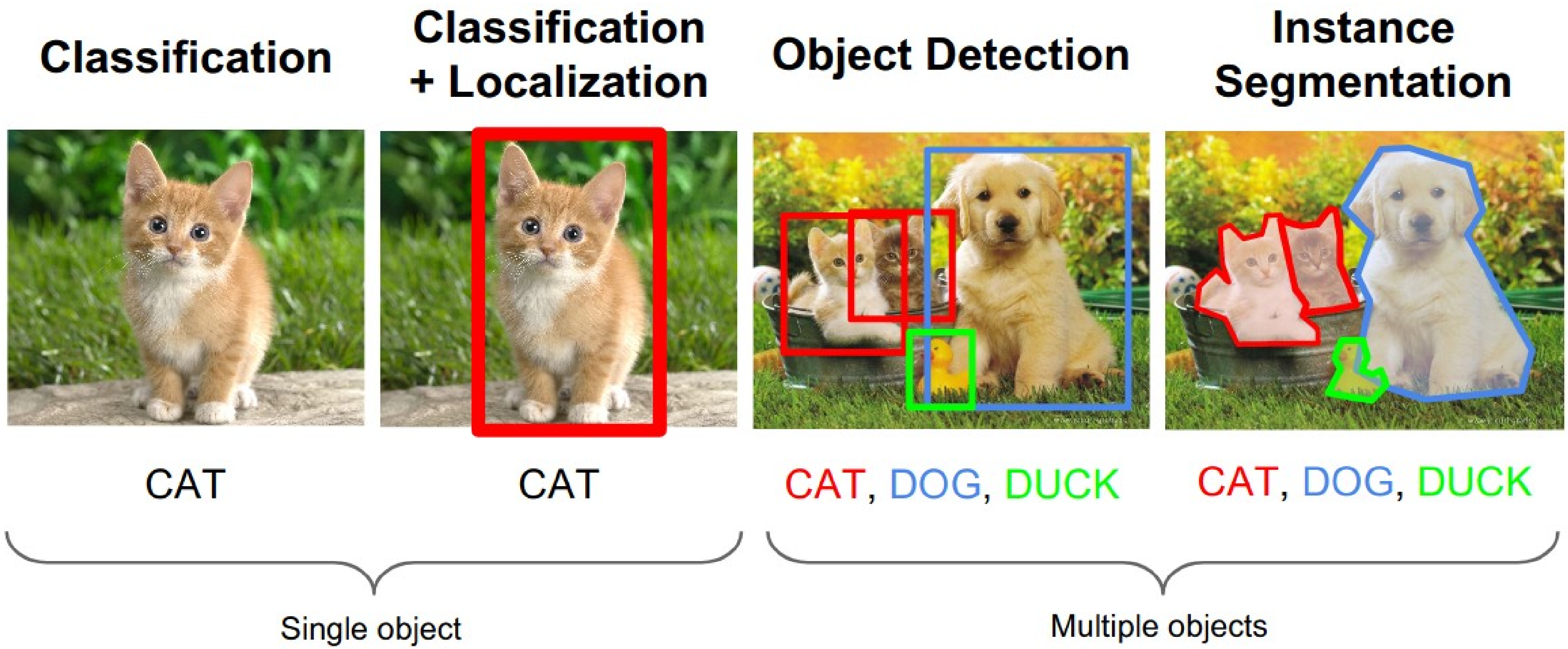
Part 3 – Detection

Object Detection

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

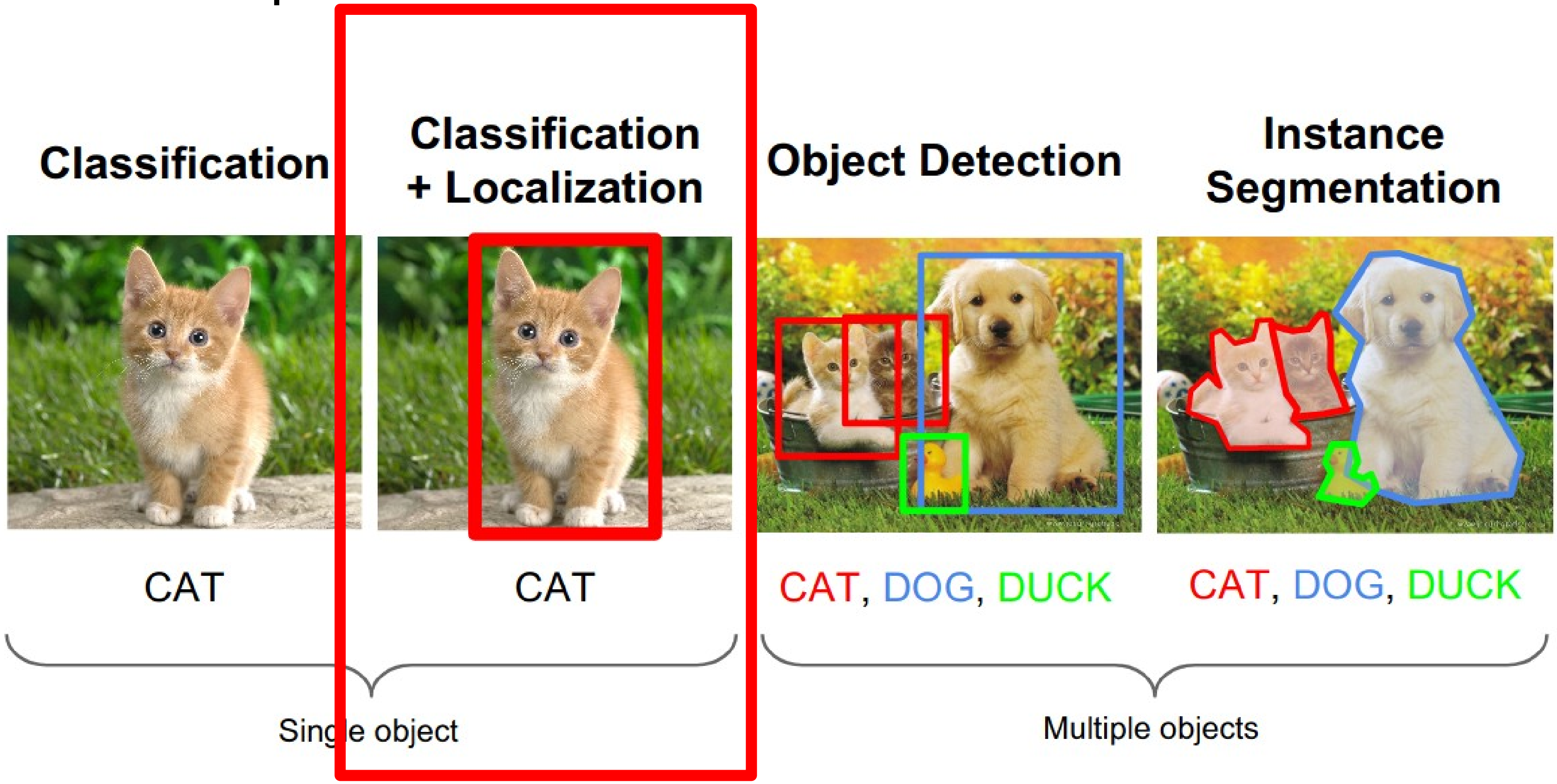
Introduction

- Different problems for vision



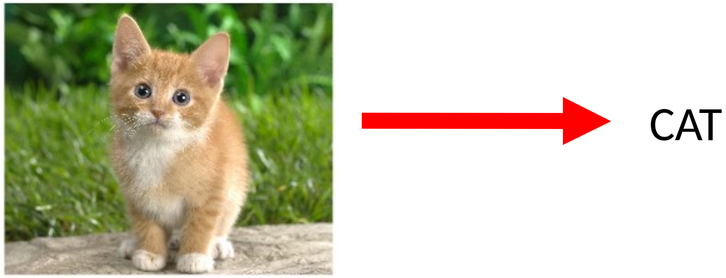
Introduction

- Different problems for vision

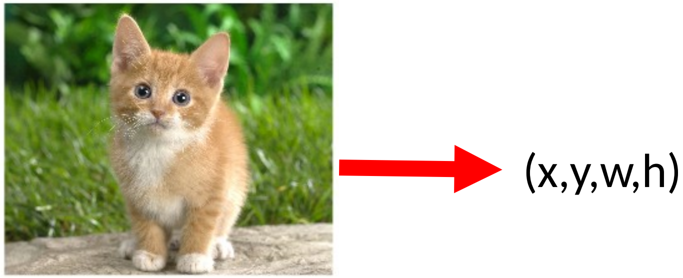


Introduction

- Classification + Localization: Task



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

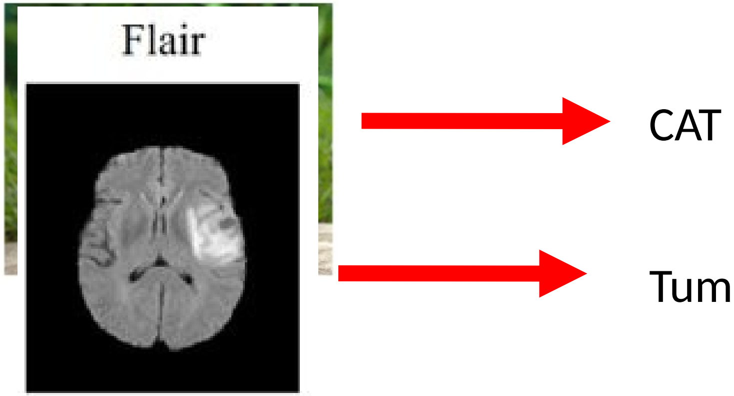


- Localization:
  - Input: Image
  - Output: Box in the image (x, y, w, h)
  - Evaluation Metric: Intersection over Union

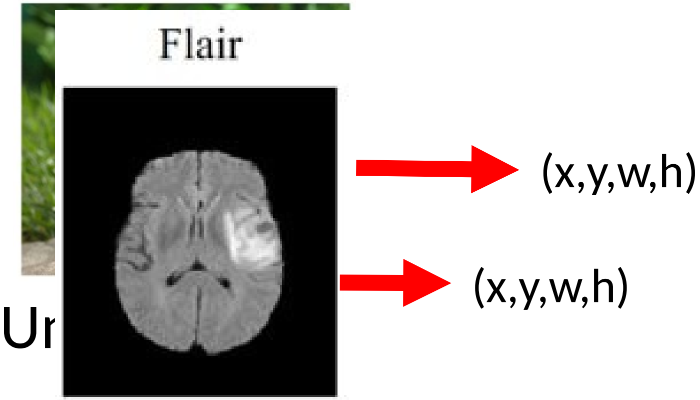
$$IoU = \frac{|A \cap B|}{|A \cup B|}$$

Introduction

- Classification + Localization: Task

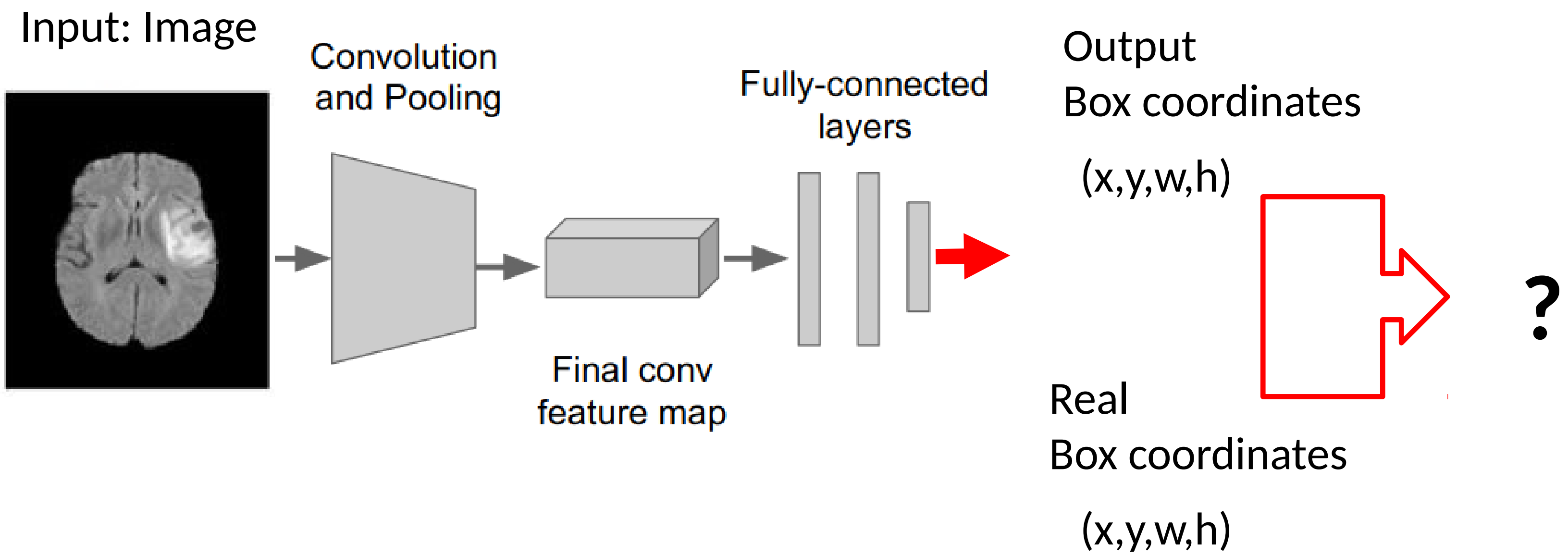


- Classification: C classes
  - Input: Image
  - Output: Class label
  - Evaluation Metric: Accuracy



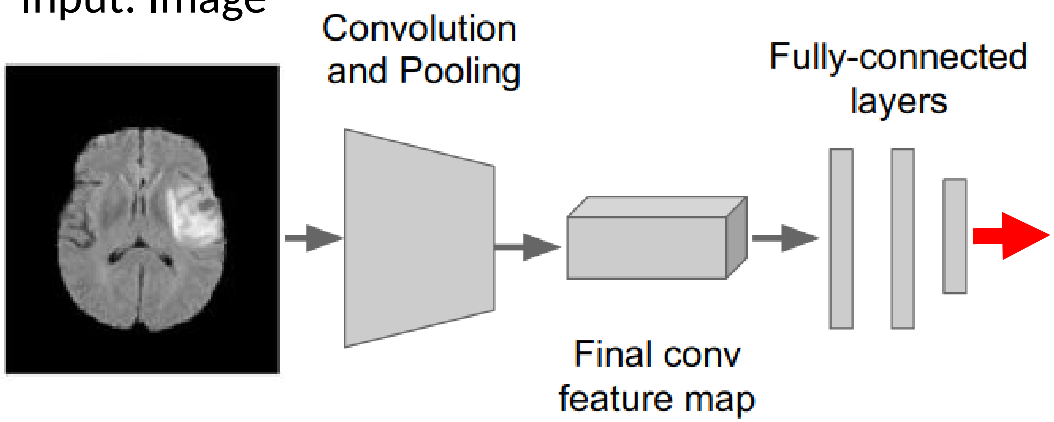
- Localization:
  - Input: Image
  - Output: Box in the image (x, y, w, h)
  - Evaluation Metric: Intersection over Ur

Localization



Idea: Localization as Regression

Input: Image



Output
Box coordinates

(x,y,w,h)

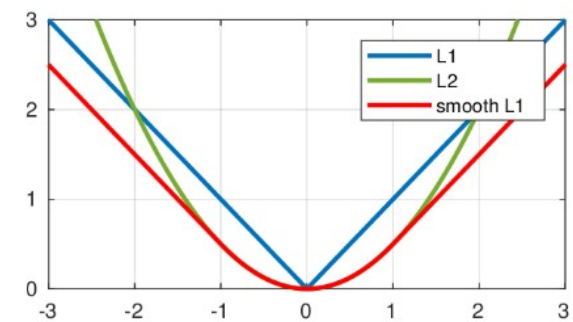
L2 Loss

$$\ell(y, f(x)) = (y - f(x))^2$$

Real
Box coordinates

(x,y,w,h)

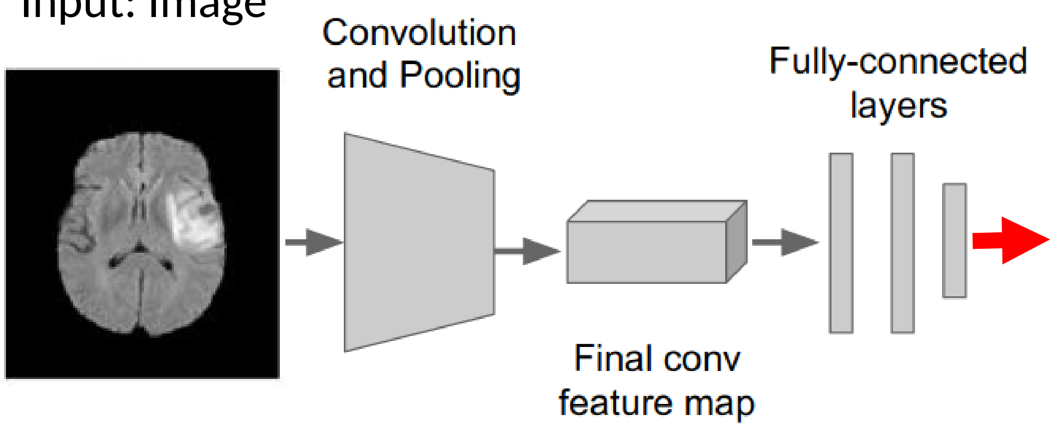
$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$



Simple Idea for Classification & Localization

- **Step 1:** Train (or download) a classification model (AlexNet, VGG, DenseNet, …)

Input: Image

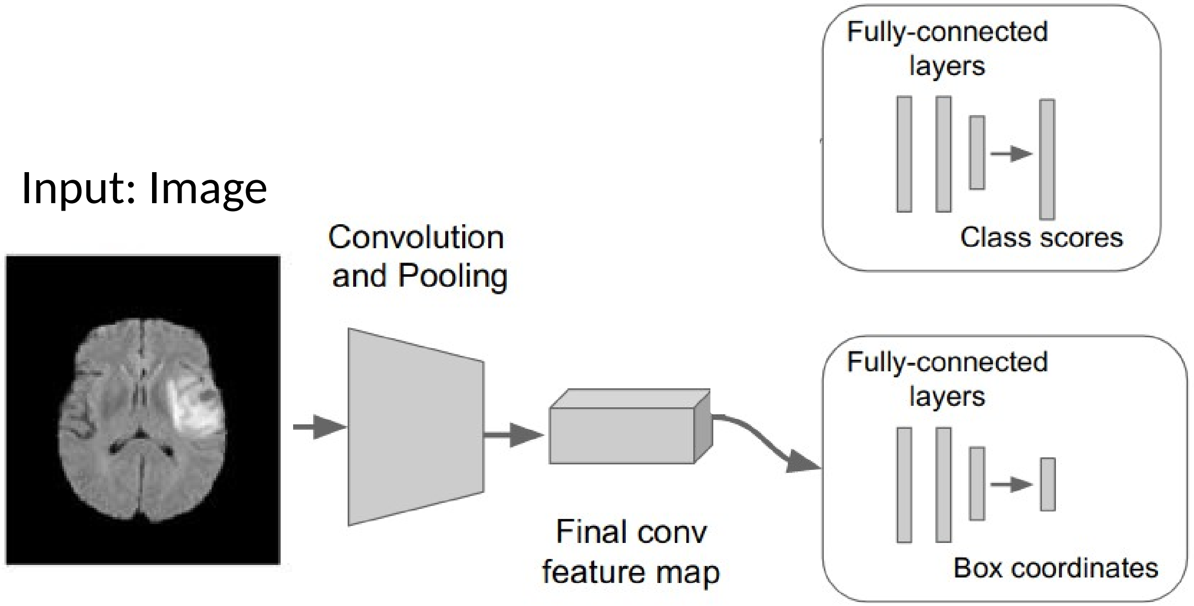


Binary Cross Entropy

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

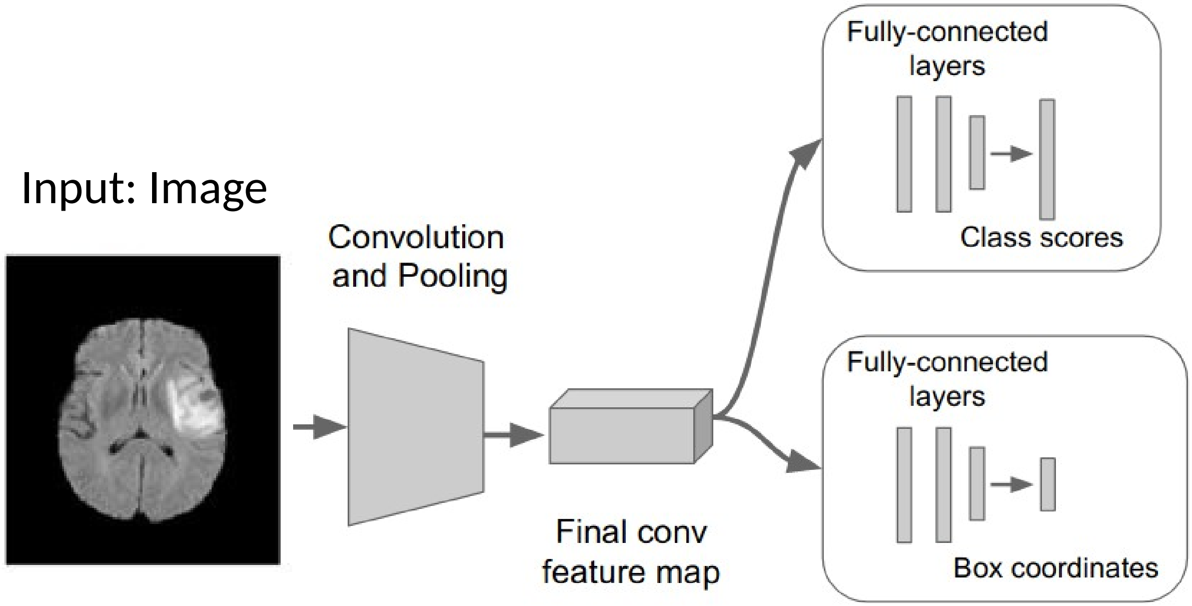
Simple Idea for Classification & Localization

- **Step 1:** Train (or download) a classification model (AlexNet, VGG, DenseNet, …)
- **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

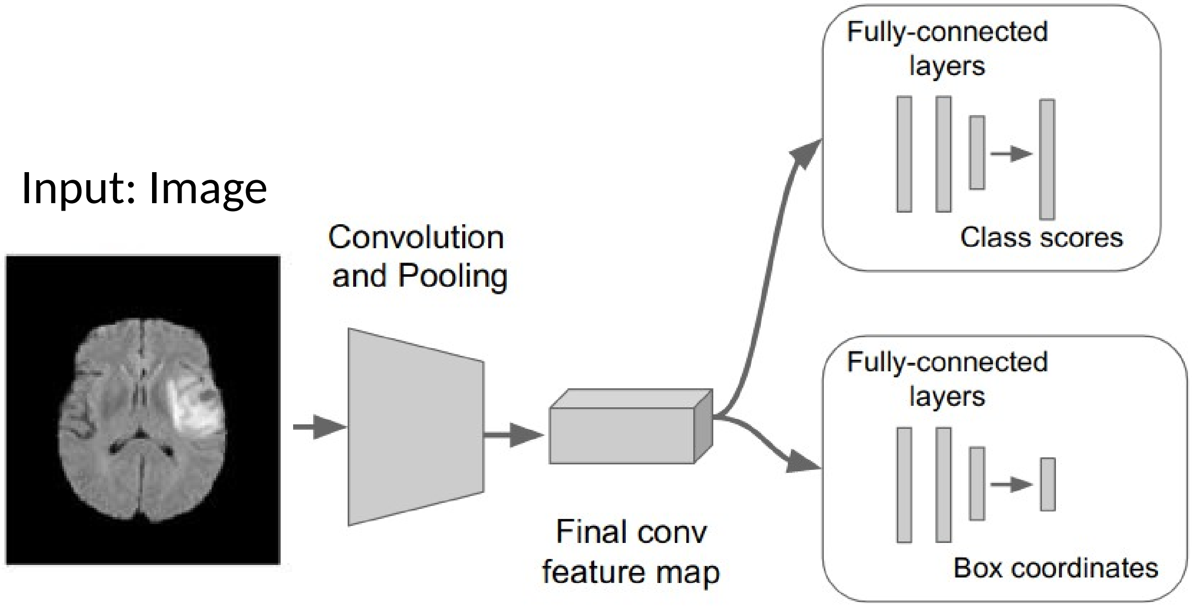
- **Step 1:** Train (or download) a classification model (AlexNet, VGG, DenseNet, …)
- **Step 2:** Attach new fully connected "regression" to the network
- **Step 3:** Train the regression part only using the L2 loss
- **Step 4:** At test time use both branches



Simple Idea for Classification & Localization

- **Step 1:** Train (or download) a classification model (AlexNet, VGG, DenseNet, …)
- **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

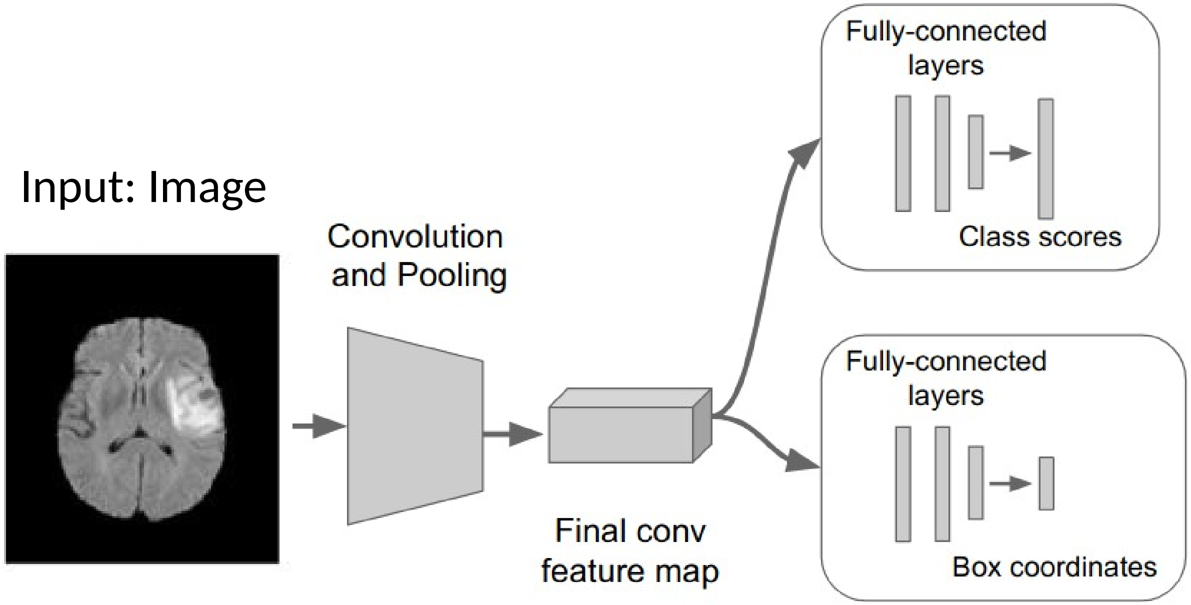
Possible choices???



Per-class vs class agnostic regresion

- Assume we are doing classification over C classes:

Input: Image

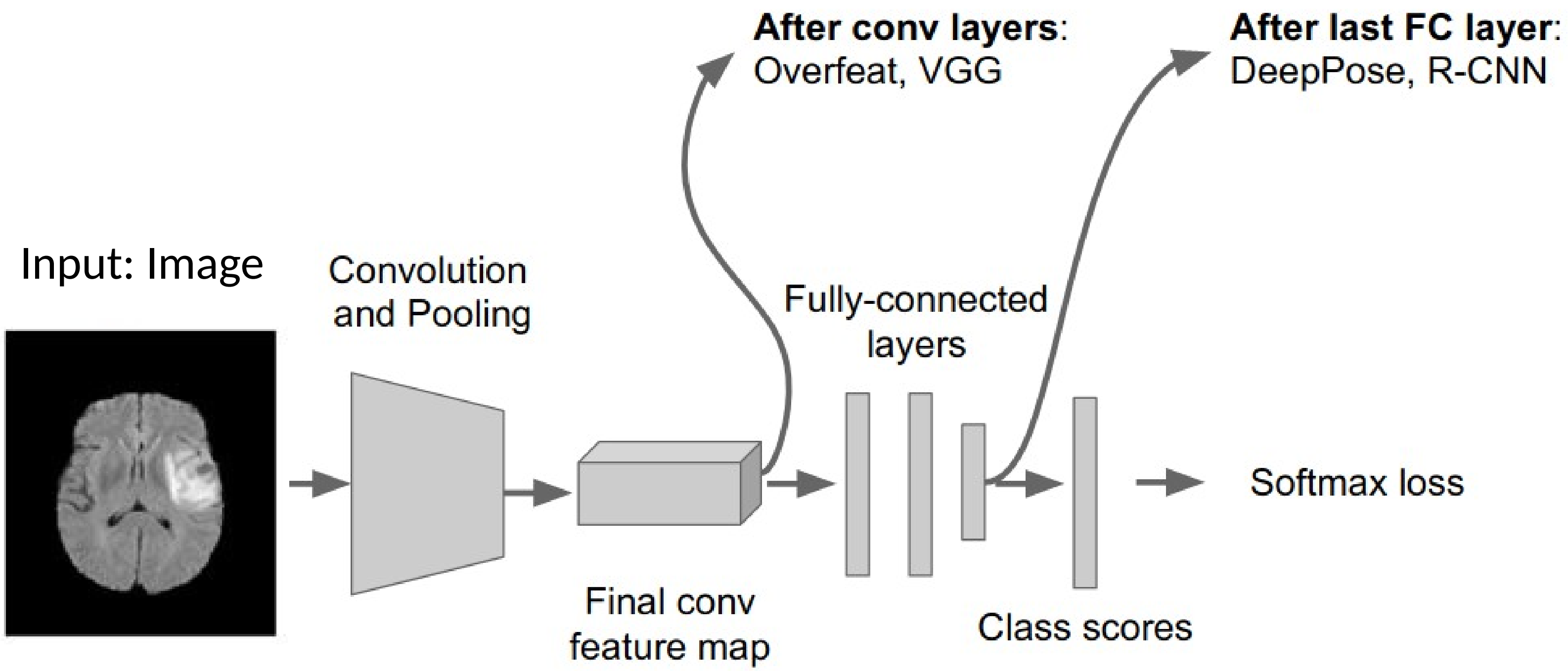


Classification Branch:
C numbers (one per class)

Class agnostic:
4 numbers (One box)

Class specific
Cx4 numbers (one box per class)

Where to attach the regression branch?



Better localization?

- Localization as Regression
    - Very simple
    - But ….

Better localization?

- Localization as Regression
  - Very simple
  - But ….

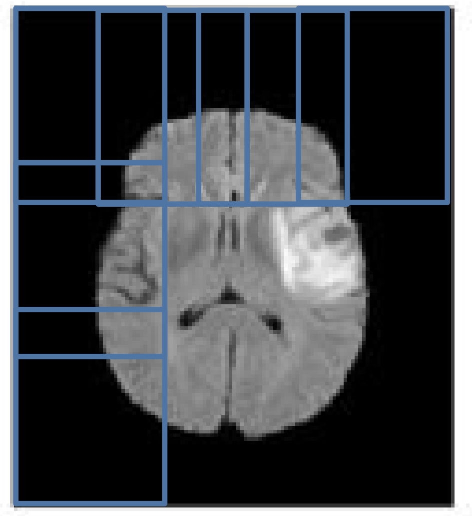
- **Idea 2: Sliding Window**
  - Run classification + localization in multiple locations on a high-resolution image
  - Convert fully-connected layers into convolutional layers for efficient computation
  - Combine classifier and regressor predictions across all scales for final prediction

Sliding Window

0.1			

Classification Scores

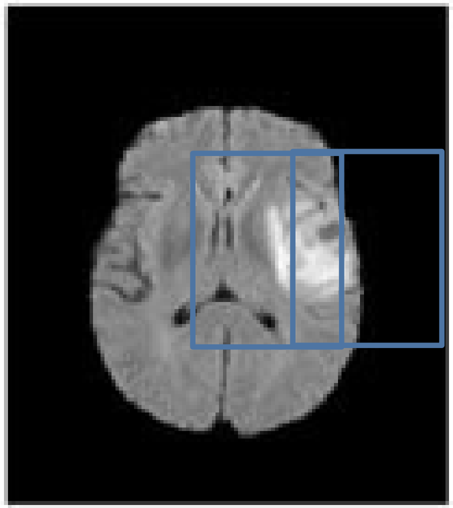
Sliding Window



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

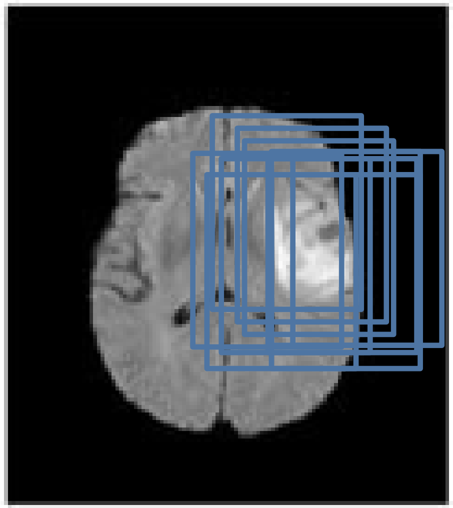
Sliding Window



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

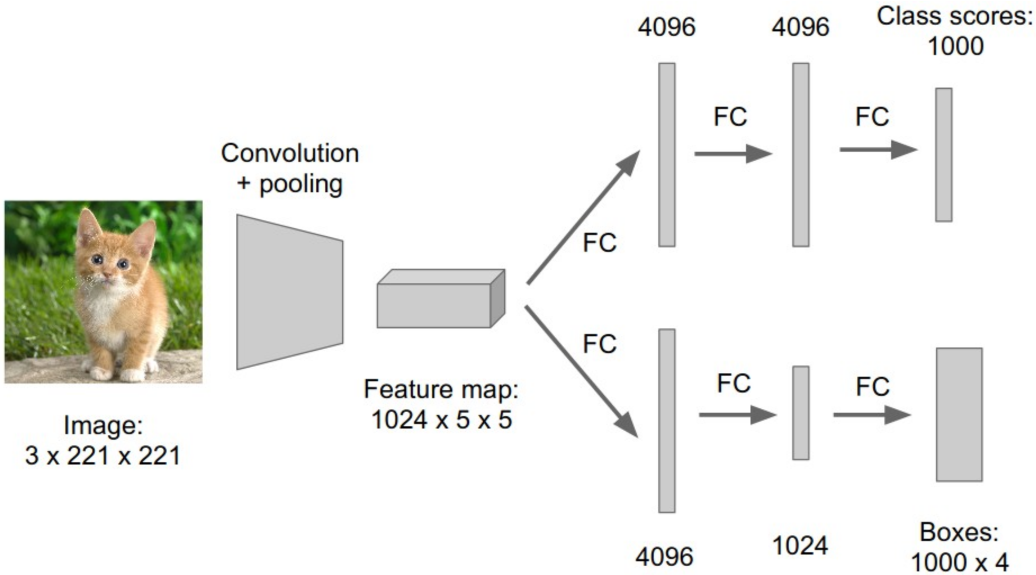
Classification Scores

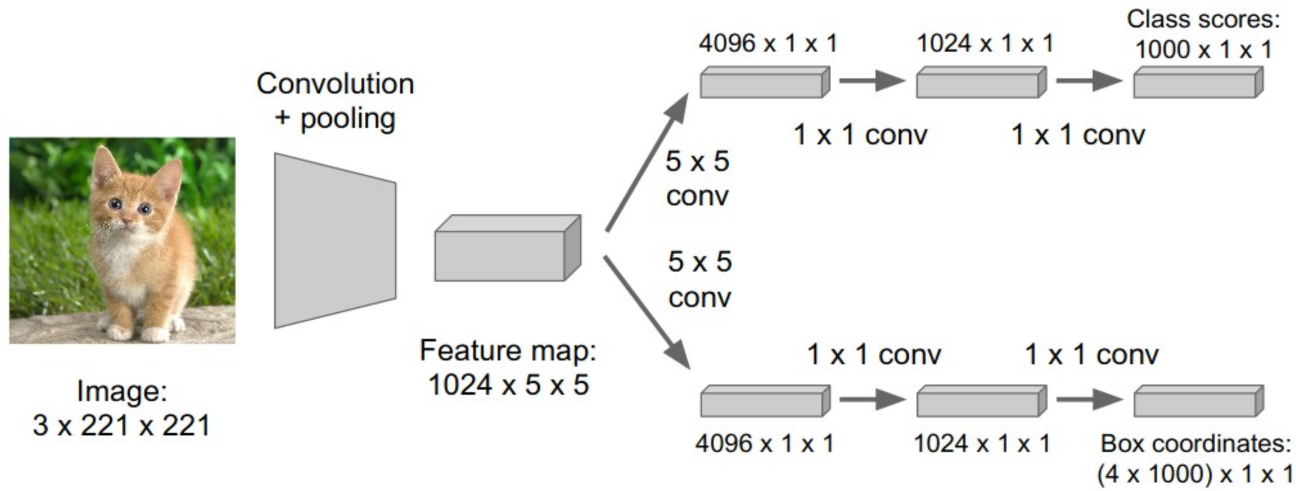
Select the proper bbox

- 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



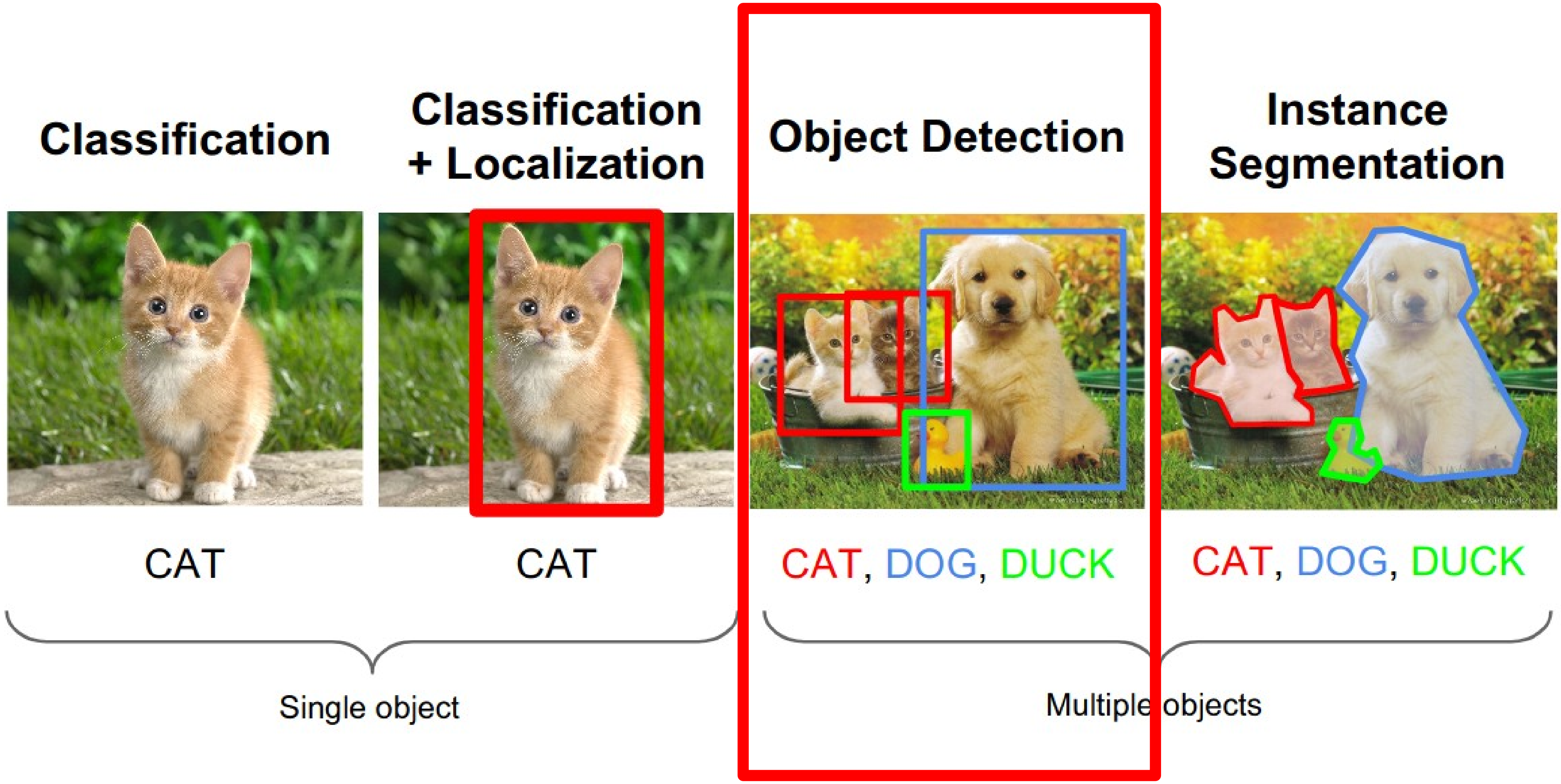


Localization & Classification

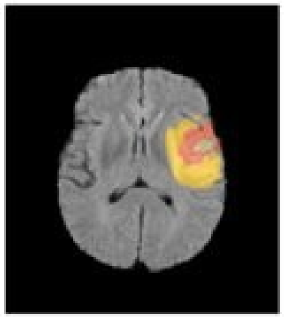
Recap

- 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

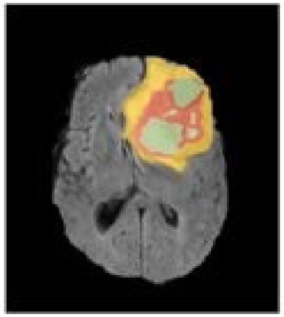


Object detection as Regression?



- 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)

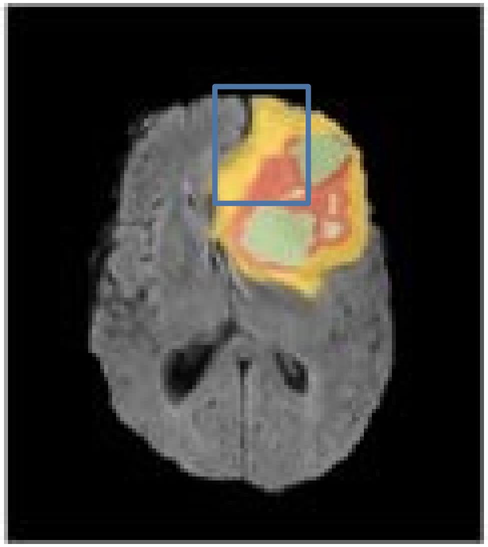
16 parameters to regress

• • •

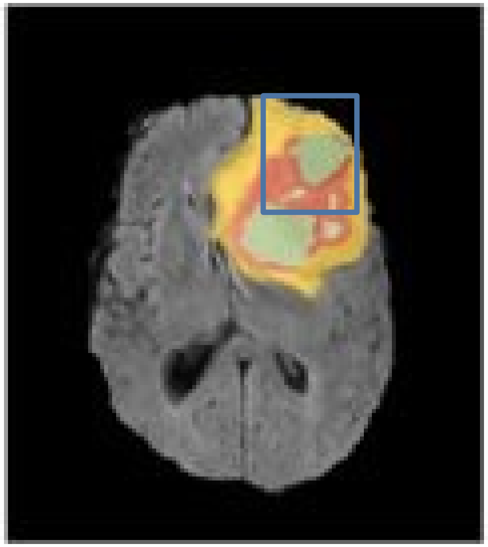
• • •

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?

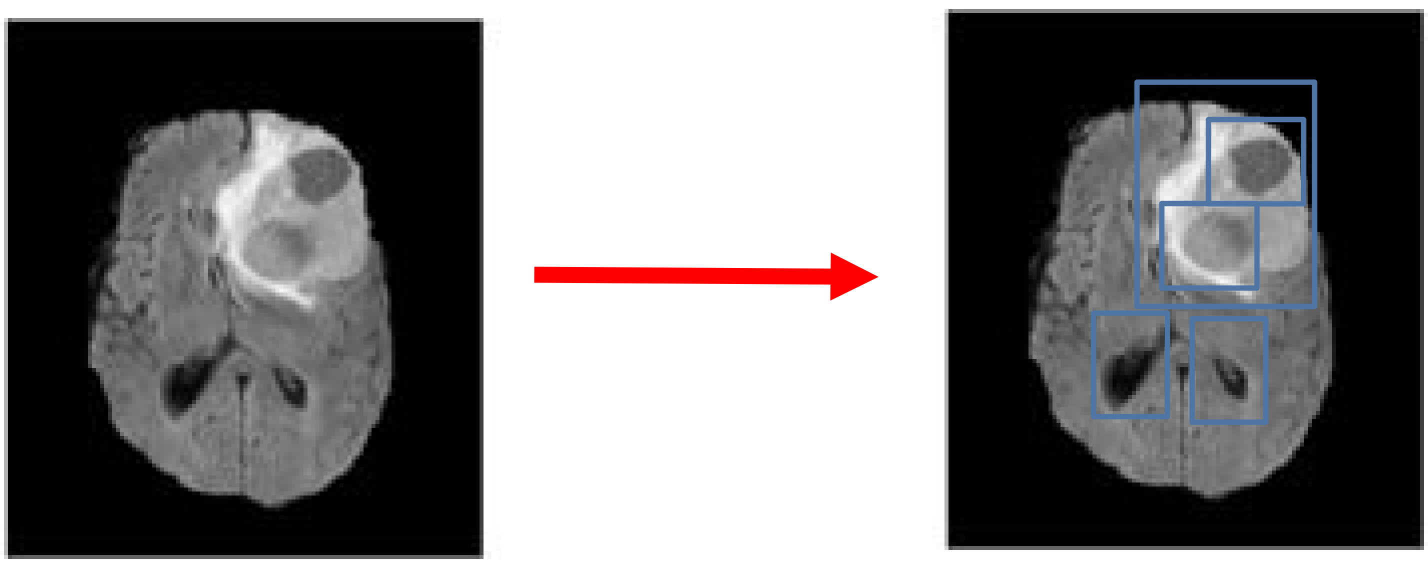
- **Problem:** Need to test many positions and scales
  - Apply CNN to every possible crop of the image and take the class

- **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 contain objects (e.g. interesting image areas)

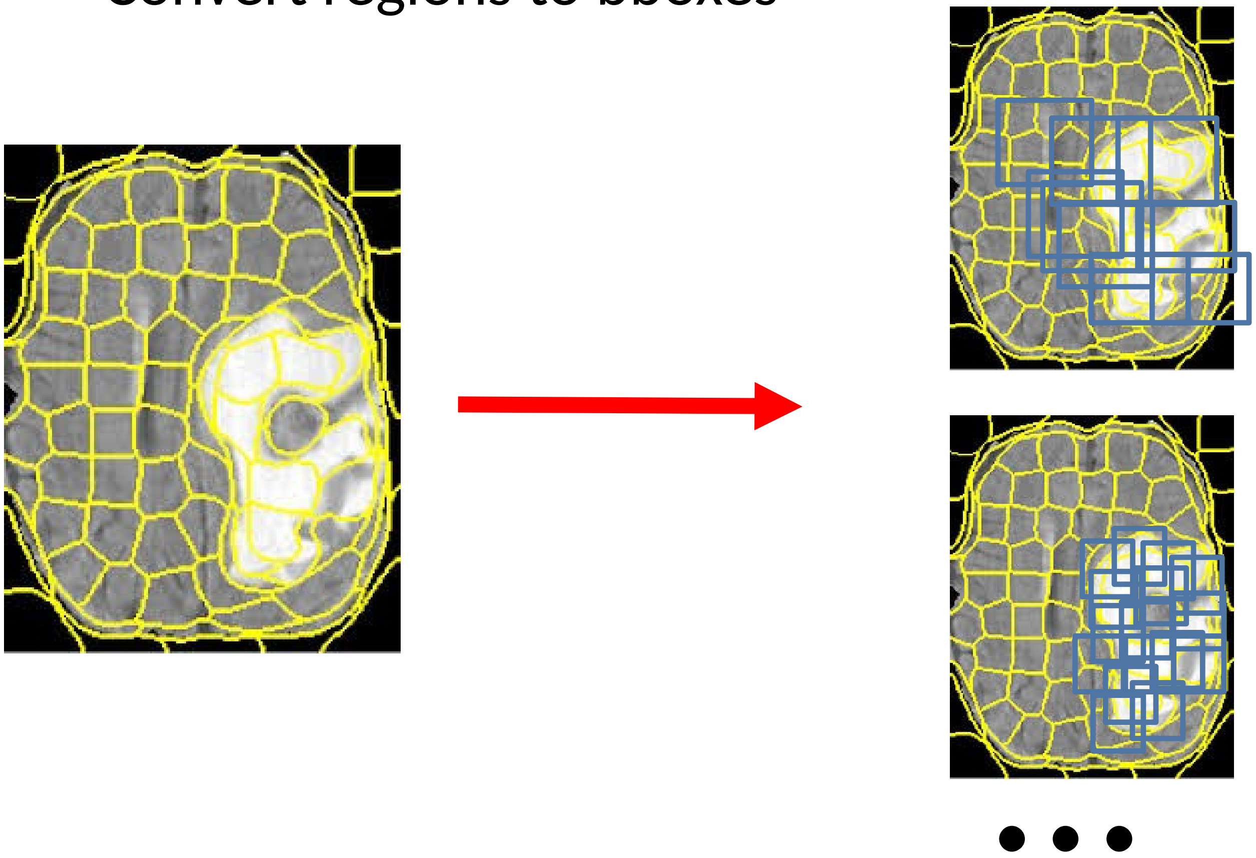
Region Proposals

- Find blobby image regions that are likely to contain objects
- Class independent 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

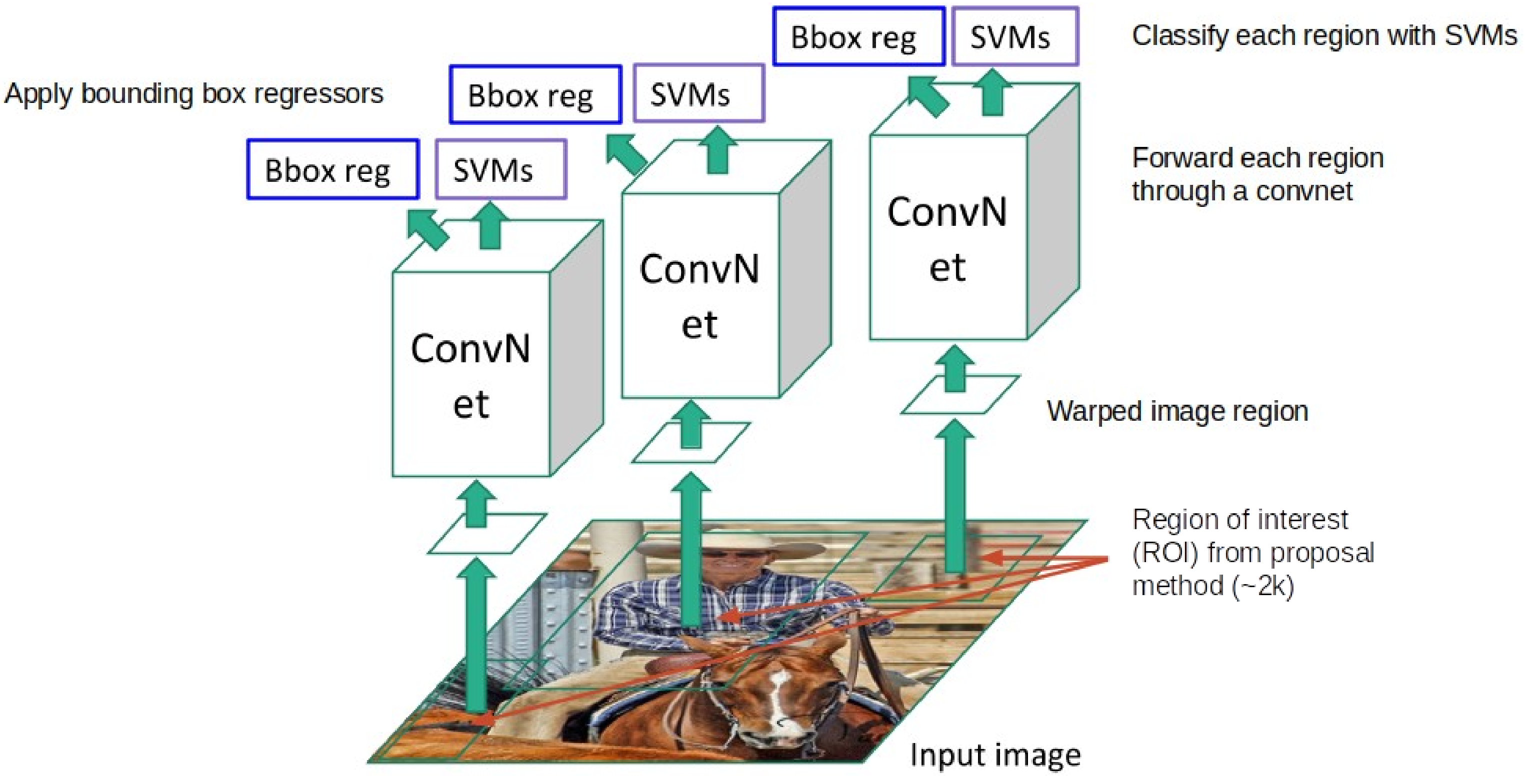
Region Proposals

- 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



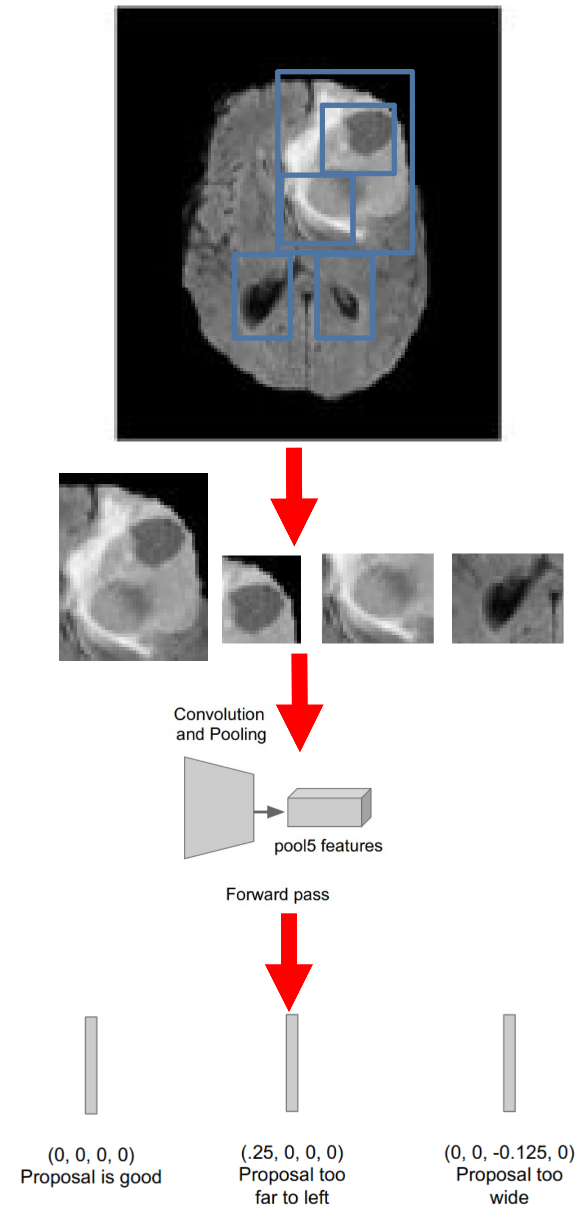
Regression Loss

$$\mathcal{L}_{\text{reg}} = \sum_{i \in \{x,y,w,h\}} (t_i - d_i(\mathbf{p}))^2 + \lambda \|\mathbf{w}\|^2$$

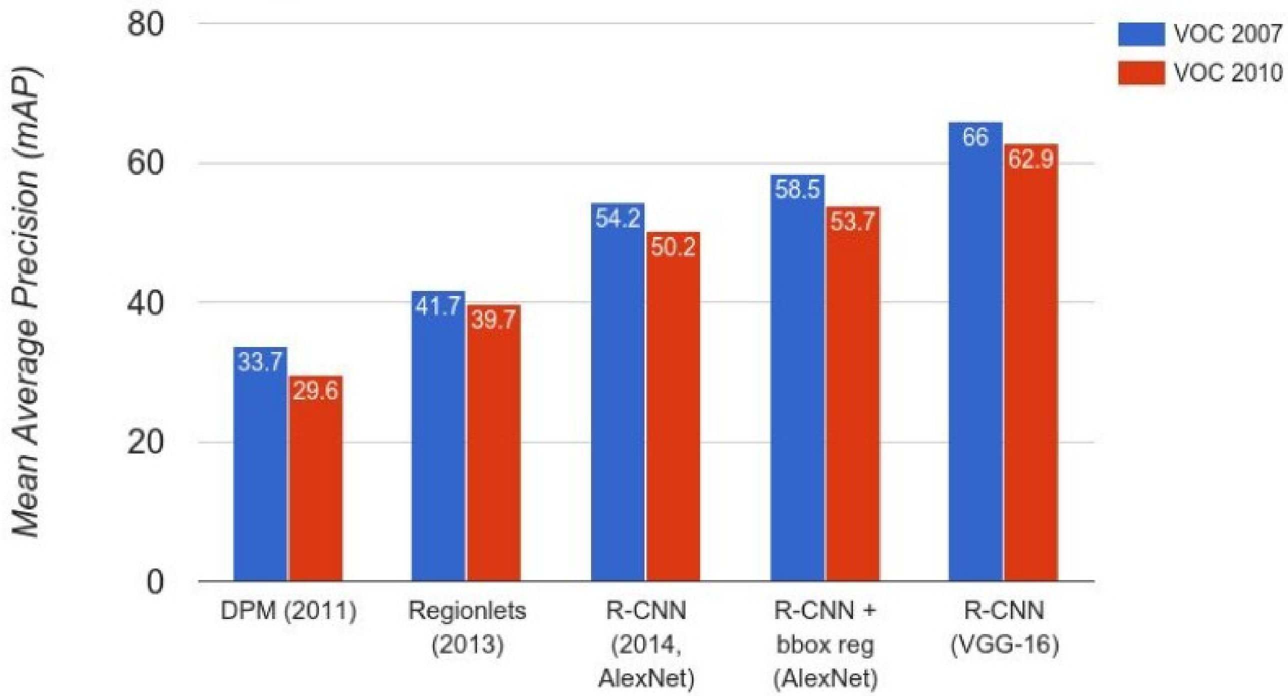
R. Girshick et al. CVPR 2014

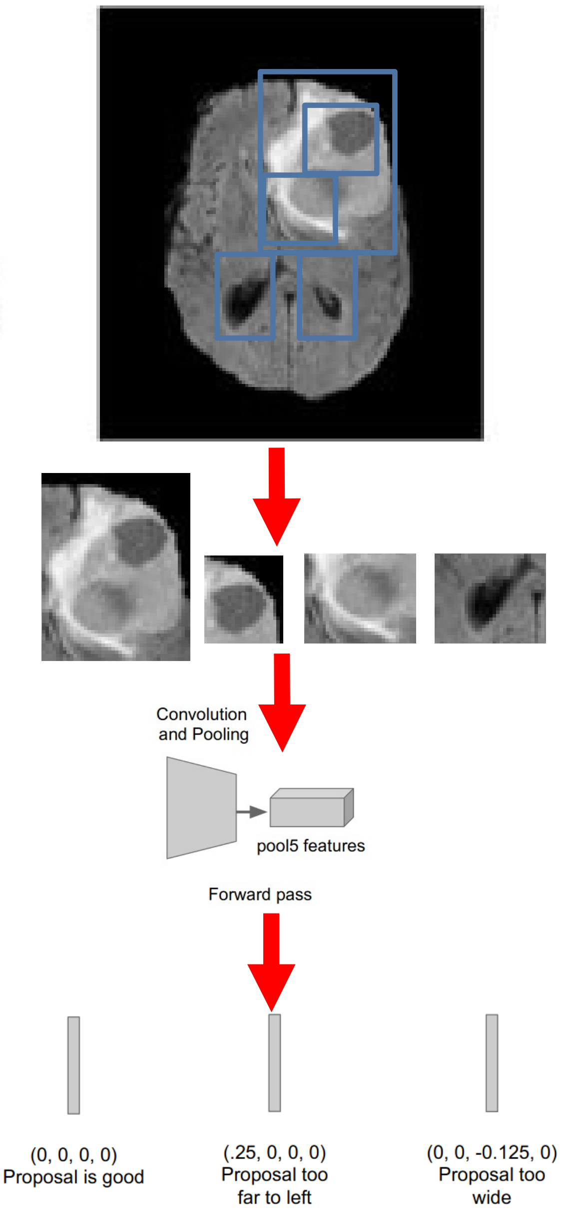
R-CNN – Recipe

- Step 1: Train a classification model for ImageNet
- 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 5th layer
- Step 4: Train one binary SVM per class to classify region features
  - Hard negative mining
- 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

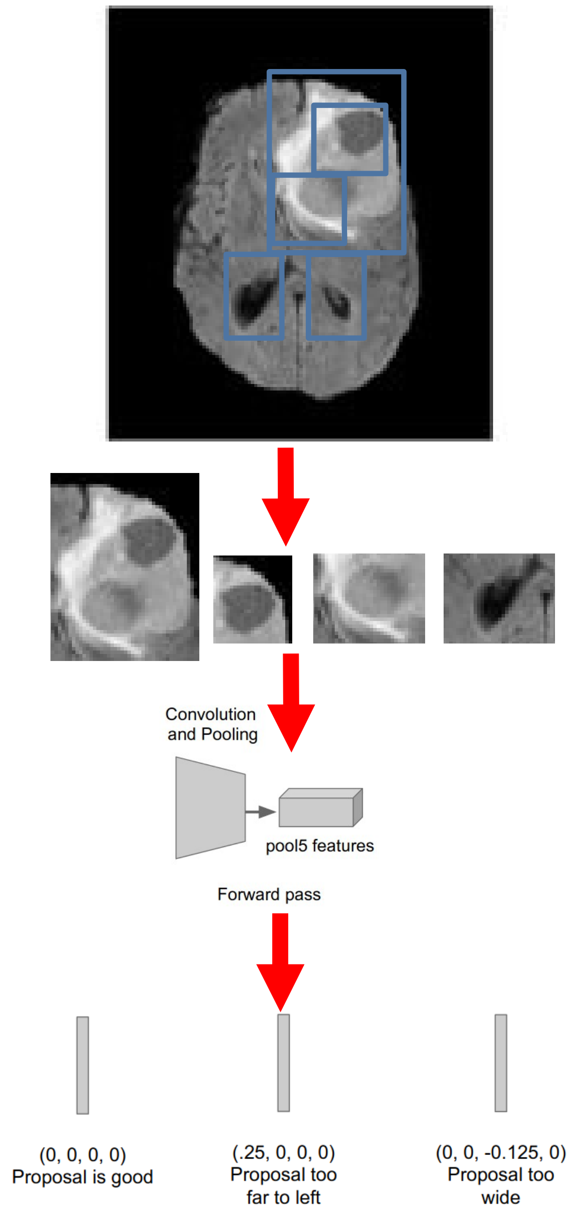




Wang et al, ICCV 2013

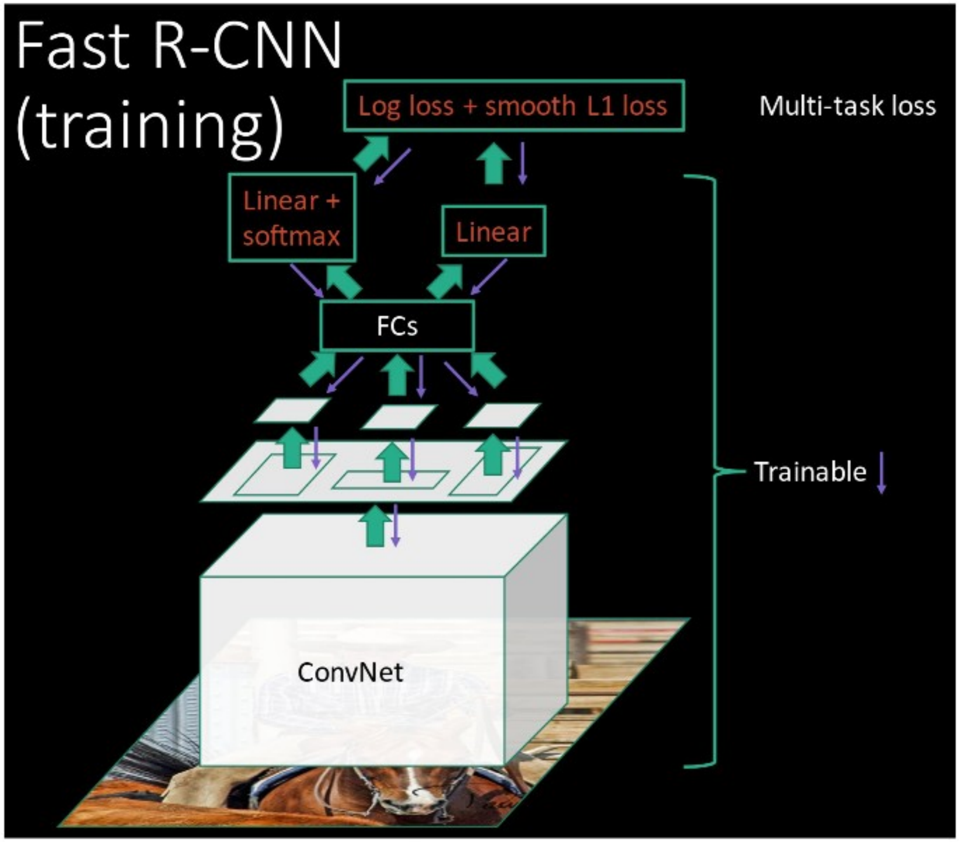
R-CNN – Problems

- 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)
- Trainins is slow, takes a lot of disk space
- Inference is slow
- Complex multistage training pipeline



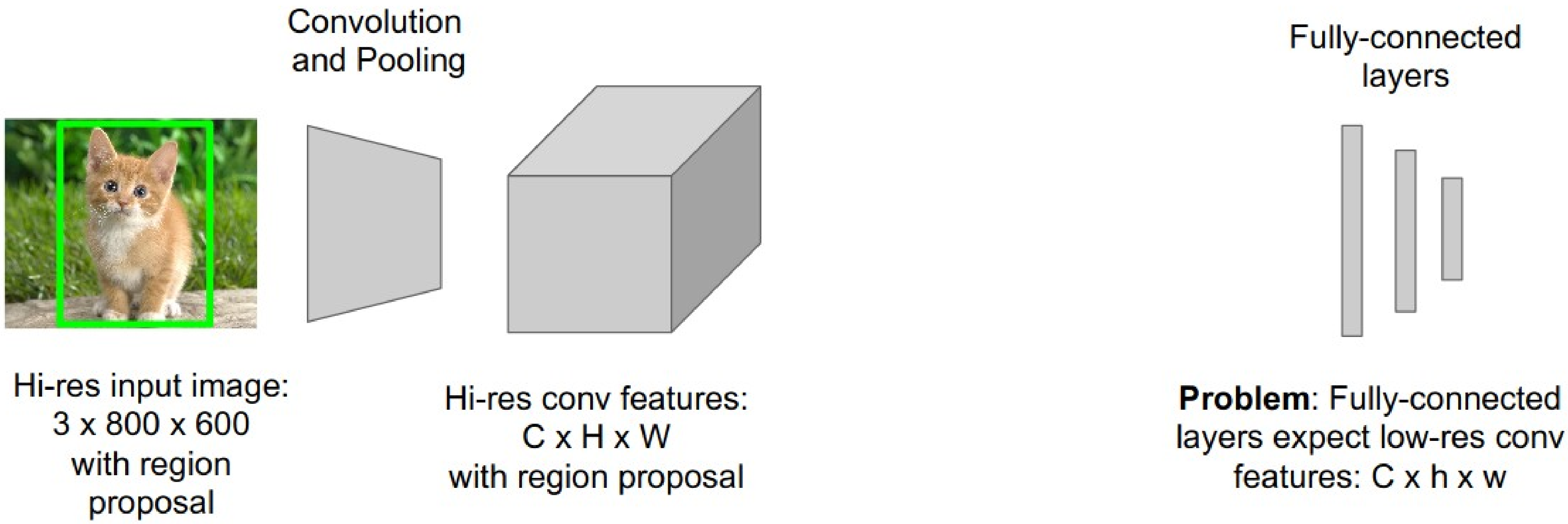
Fast R-CNN

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

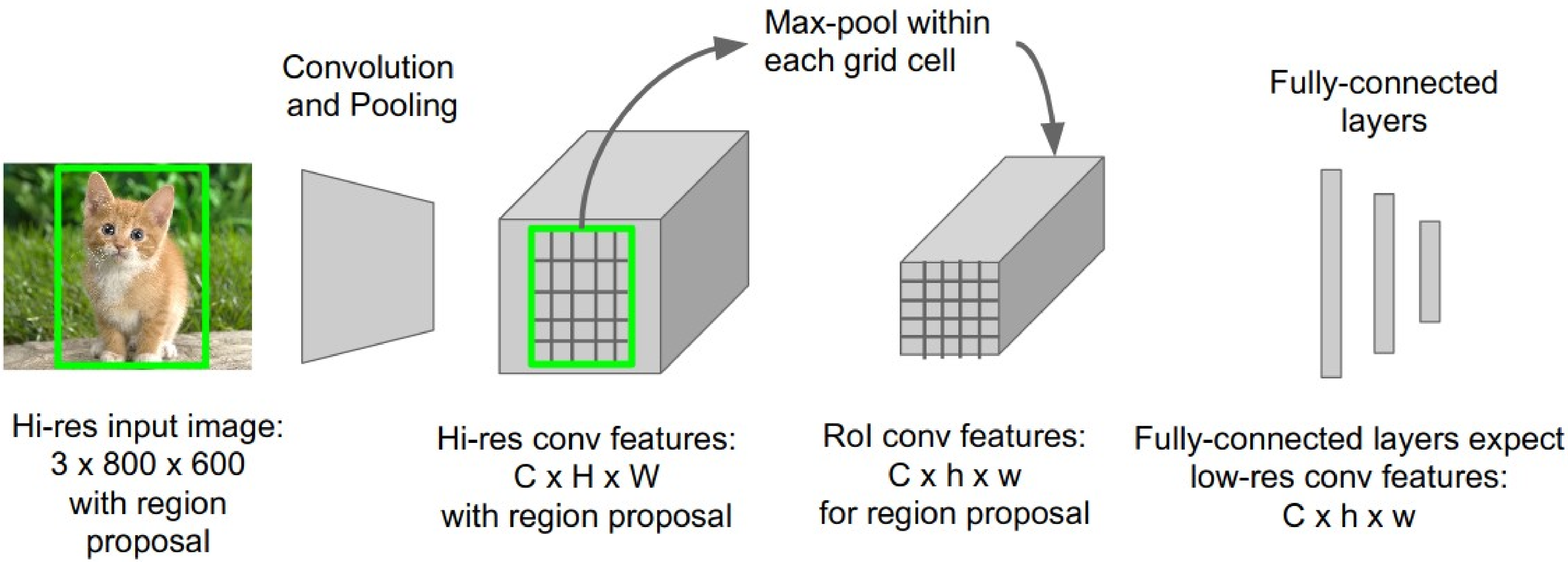


R. Girshick et al. ICCV 2015

Fast R-CNN: Region of Interest Pooling



Fast R-CNN: Region of Interest Pooling



- Train is enabled as all the process are differentiable!

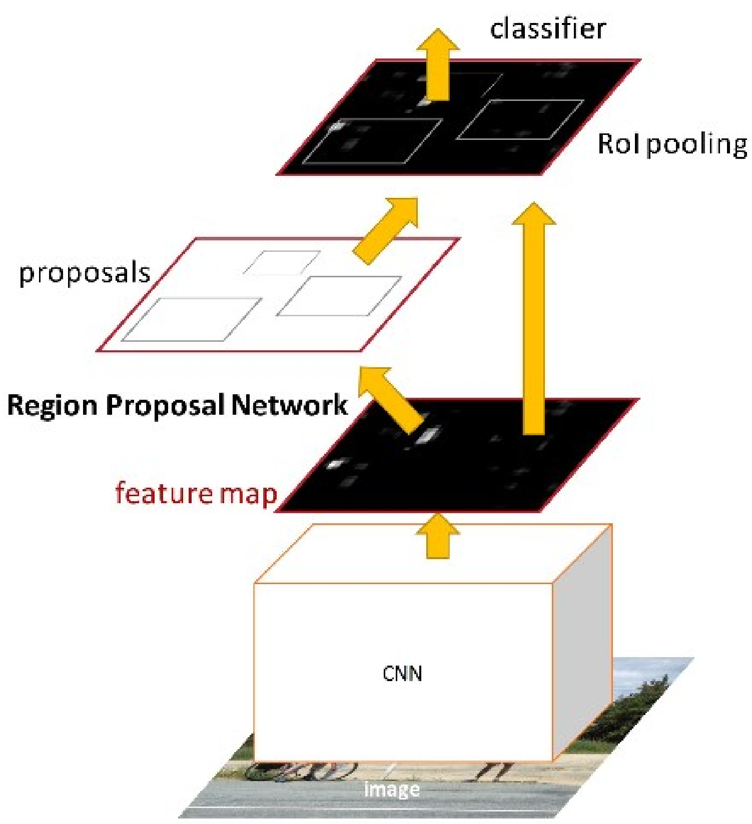
R-CNN vs Fast R-CNN

		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

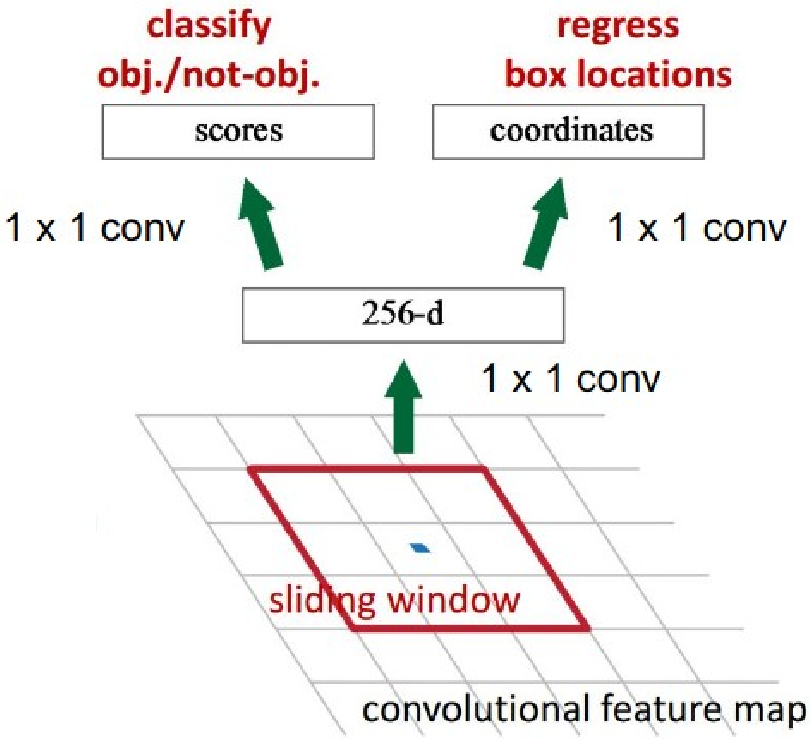
Faster R-CNN



Ren et al. NeurIPS 2015

- 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.

Faster R-CNN: Region Proposal Network



Slide a small window on the feature map

Build a small network for:

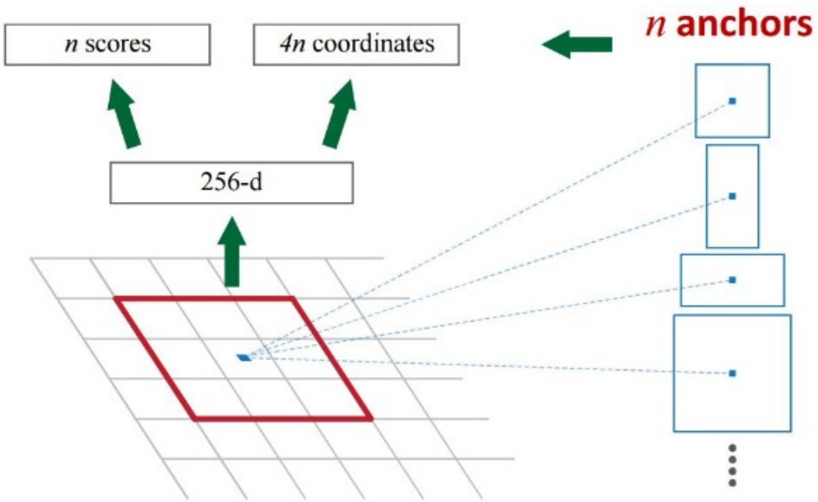
- Classifying object or not-object,
- Regressing bbox locations

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.

Ren et al. NeurIPS 2015

Faster R-CNN: Region Proposal Network



Use N anchor boxes at each location

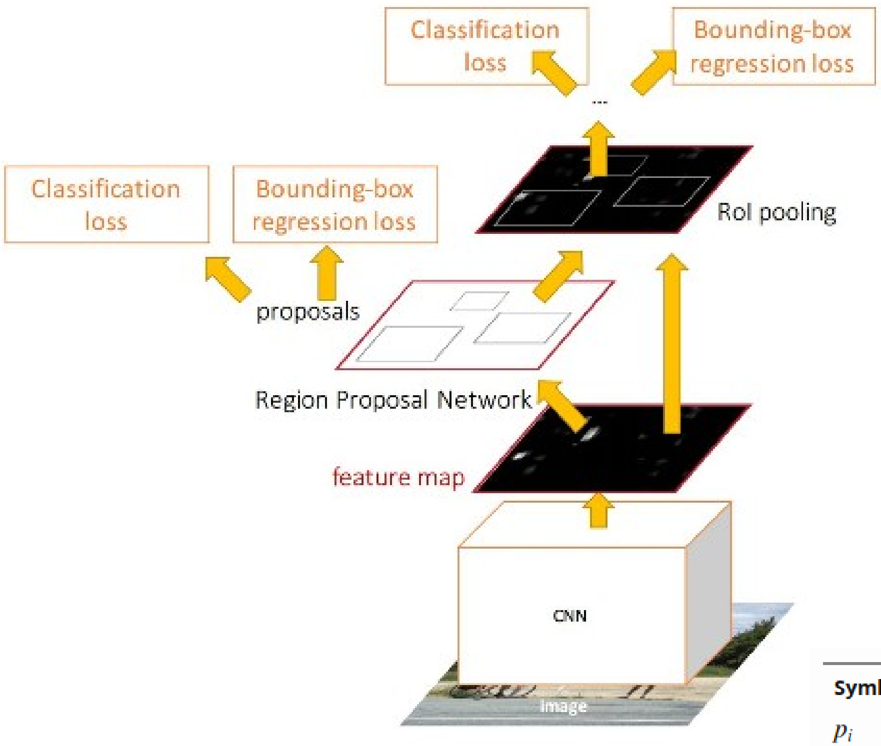
Anchors are translation invariant: use the same ones at every location

Regression gives offsets from anchor boxes

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

Ren et al. NeurIPS 2015

Faster R-CNN



Joint training:

One network, four losses

- RPN classification (anchor good/bad)
- RPN regression (anchor → proposal)

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*).$$

$$t_{\mathrm{x}}^* = (x^* - x_{\mathrm{a}})/w_{\mathrm{a}}, \quad t_{\mathrm{y}}^* = (y^* - y_{\mathrm{a}})/h_{\mathrm{a}},$$

$$t_{\mathrm{w}}^* = \log(w^*/w_{\mathrm{a}}), \quad t_{\mathrm{h}}^* = \log(h^*/h_{\mathrm{a}}),$$

- Fast R-CNN classification (over classes)
- Fast R-CNN regression (proposal → box)

Symbol	Explanation
p_i	Predicted probability of anchor i being an object.
p_i^*	Ground truth label (binary) of whether anchor i is an object.
t_i	Predicted four parameterized coordinates.
t_i^*	Ground truth coordinates.
N_{cls}	Normalization term, set to be mini-batch size (~256) in the paper.
N_{box}	Normalization term, set to the number of anchor locations (~2400) in the paper.
λ	A balancing parameter, set to be ~10 in the paper (so that both $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{box}}$ terms are roughly equally weighted).

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

More networks for Object Detection

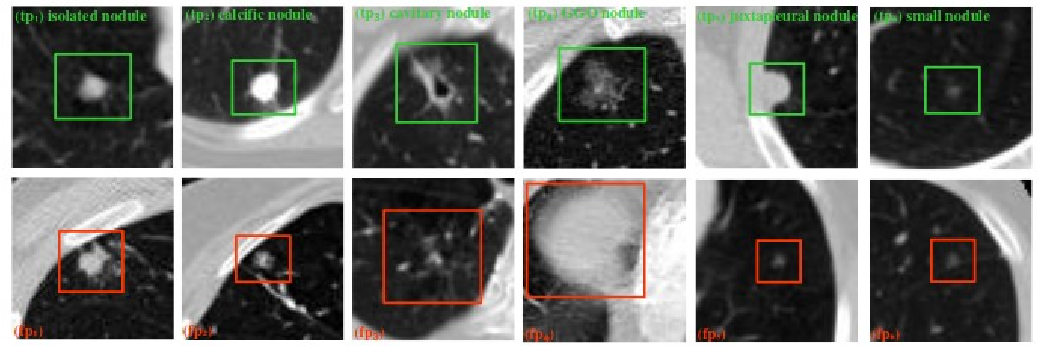
- Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection", arXiv:1506.02640
- Johnson et al., “DenseCap: Fully Convolutional Localization Networks for Dense Captioning”, CVPR 2016
- Lin et al., "Focal Loss for Dense Object Detection", ICCV 2017
- Berman M. et al., "The Lovász-Softmax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

Good but what about medical imaging?

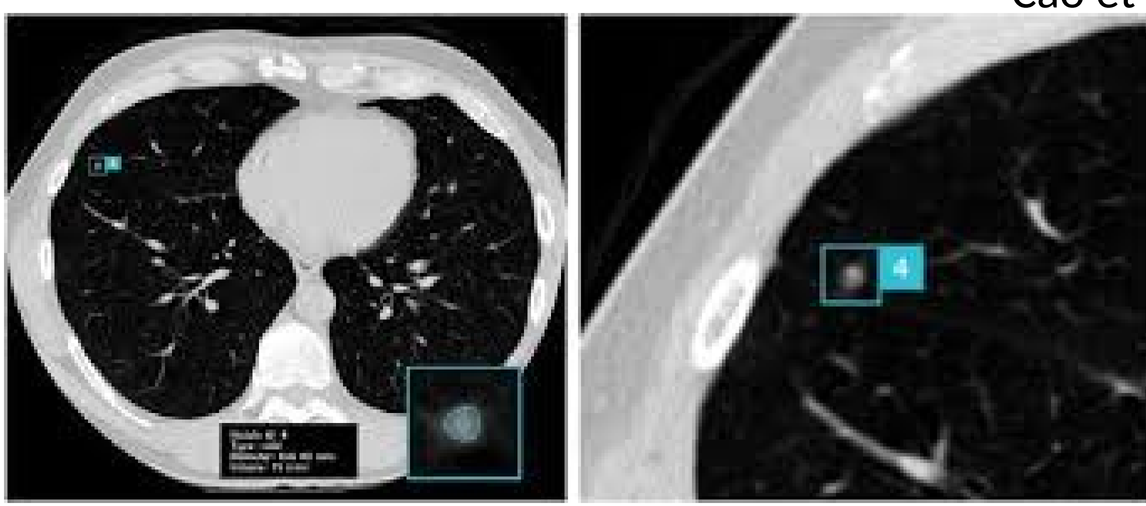
- Problems?

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.



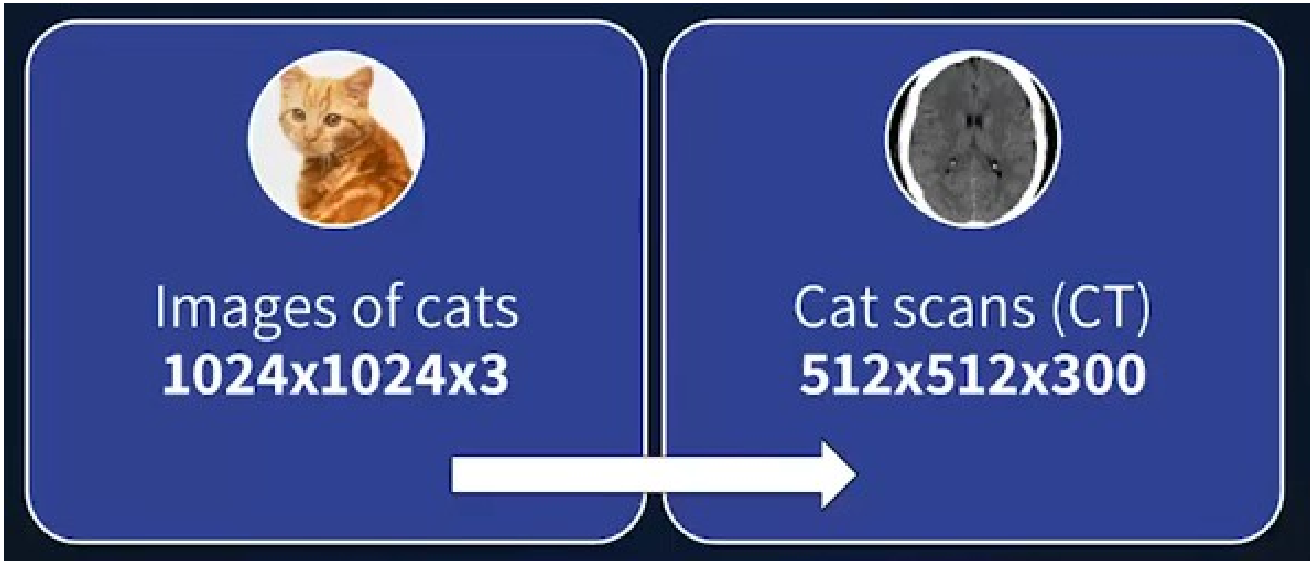
Cao et al., ArXiv 2019



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

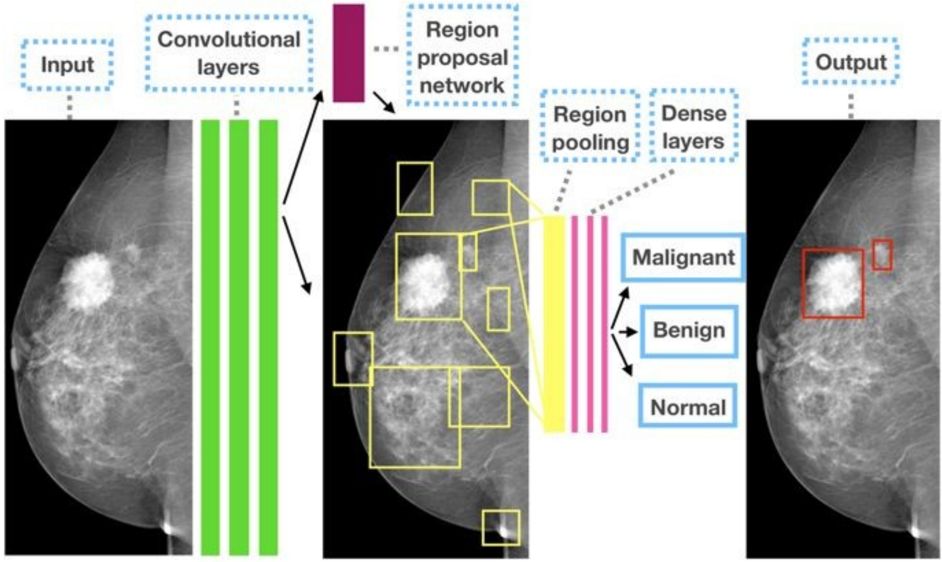


Good but what about medical imaging?

- 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

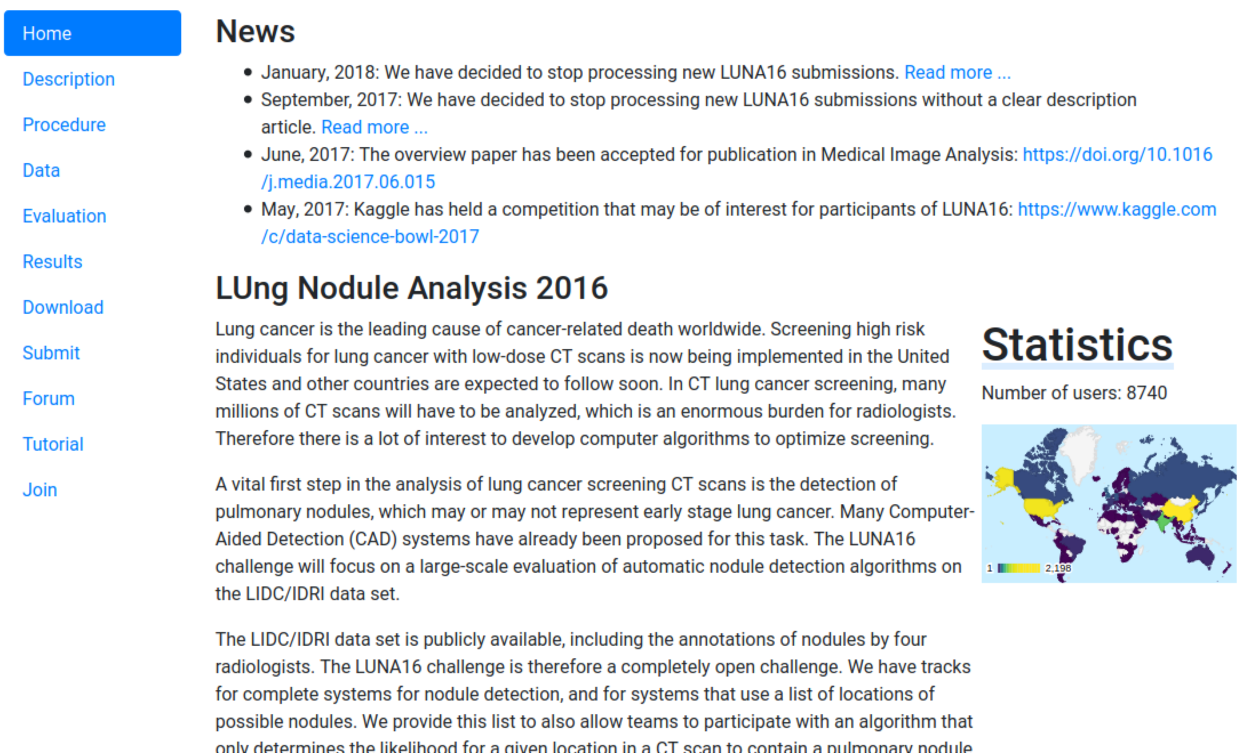
Scientific Reports 8, Article number: 4165 (2018) | Cite this article

15k Accesses | 56 Citations | 49 Altmetric | Metrics

Nodule detection

LUNA Dataset

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



Home
Description
Procedure
Data
Evaluation
Results
Download
Submit
Forum
Tutorial
Join

News

- January, 2018: We have decided to stop processing new LUNA16 submissions. Read more ...
- September, 2017: We have decided to stop processing new LUNA16 submissions without a clear description article. Read more ...
- June, 2017: The overview paper has been accepted for publication in Medical Image Analysis: https://doi.org/10.1016/j.media.2017.06.015
- May, 2017: Kaggle has held a competition that may be of interest for participants of LUNA16: https://www.kaggle.com/c/data-science-bowl-2017

LUng Nodule Analysis 2016

Lung cancer is the leading cause of cancer-related death worldwide. Screening high risk individuals for lung cancer with low-dose CT scans is now being implemented in the United States and other countries are expected to follow soon. In CT lung cancer screening, many millions of CT scans will have to be analyzed, which is an enormous burden for radiologists. Therefore there is a lot of interest to develop computer algorithms to optimize screening.

A vital first step in the analysis of lung cancer screening CT scans is the detection of pulmonary nodules, which may or may not represent early stage lung cancer. Many Computer-Aided Detection (CAD) systems have already been proposed for this task. The LUNA16 challenge will focus on a large-scale evaluation of automatic nodule detection algorithms on the LIDC/IDRI data set.

The LIDC/IDRI data set is publicly available, including the annotations of nodules by four radiologists. The LUNA16 challenge is therefore a completely open challenge. We have tracks for complete systems for nodule detection, and for systems that use a list of locations of possible nodules. We provide this list to also allow teams to participate with an algorithm that only determines the likelihood for a given location in a CT scan to contain a pulmonary nodule

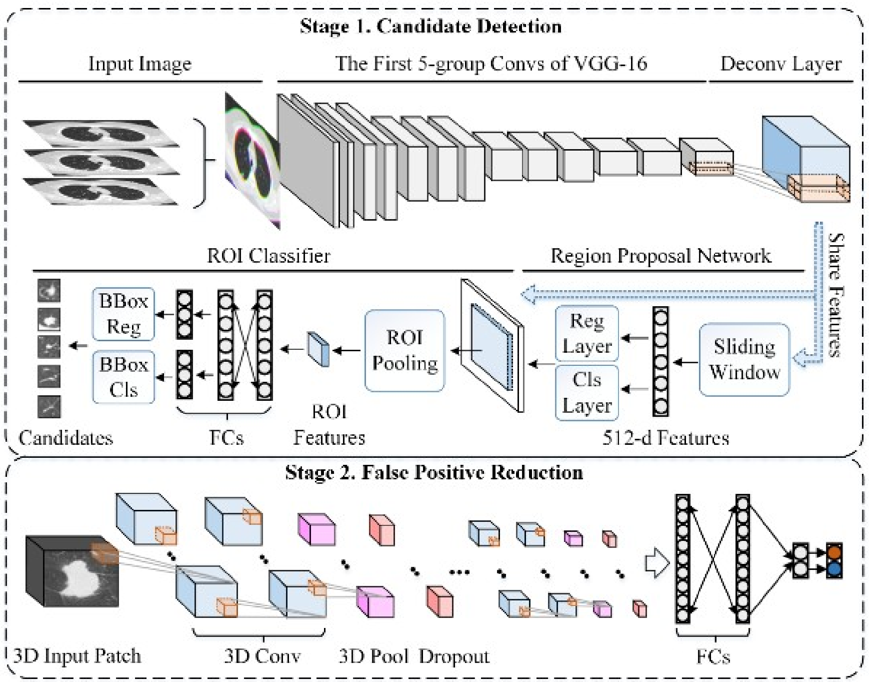
Statistics

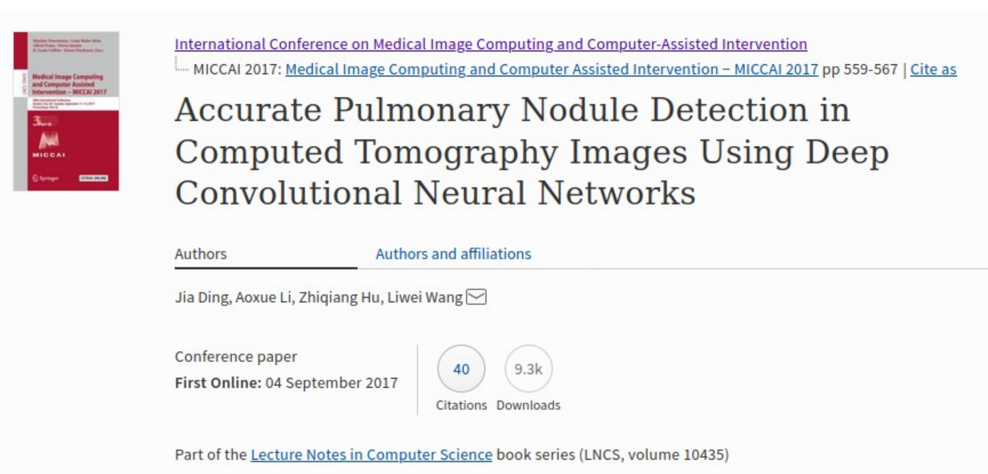
Number of users: 8740

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

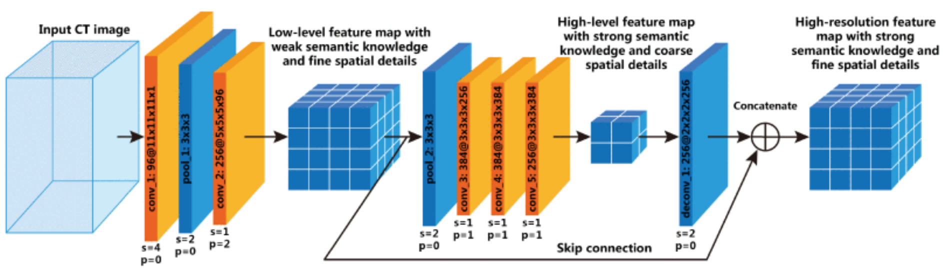
First Online: 04 September 2017

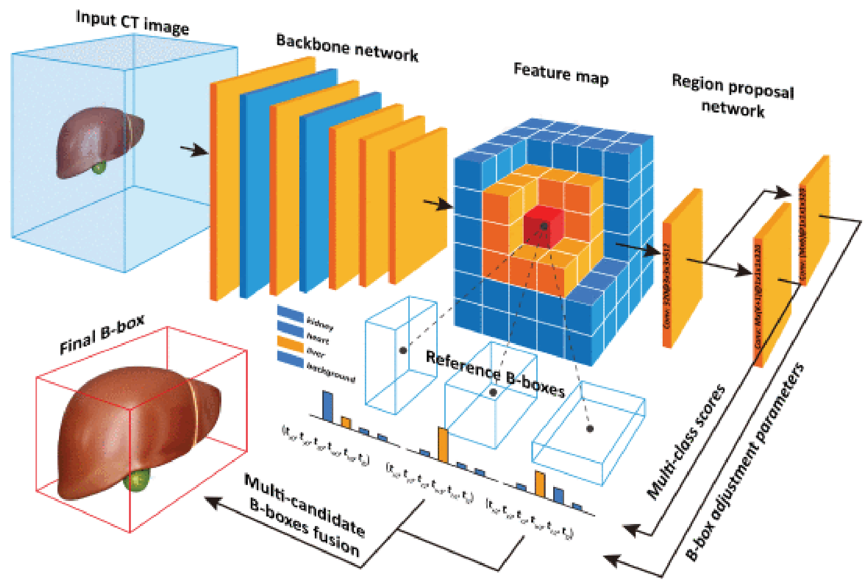
40 Citations 9.3k Downloads

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

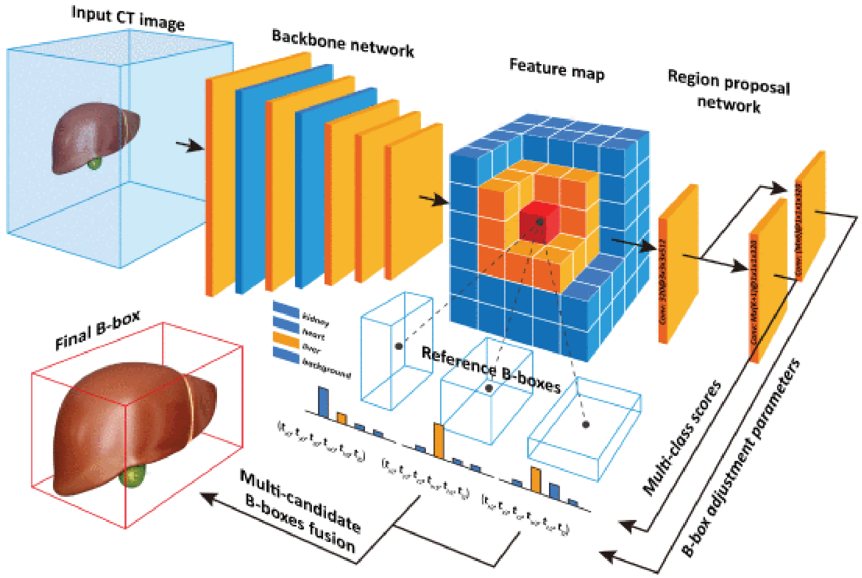
5 Author(s) Xuanang Xu ; Fugen Zhou ; Bo Liu ; Dongshan Fu ; Xiangzhi Bai View All Authors

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_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

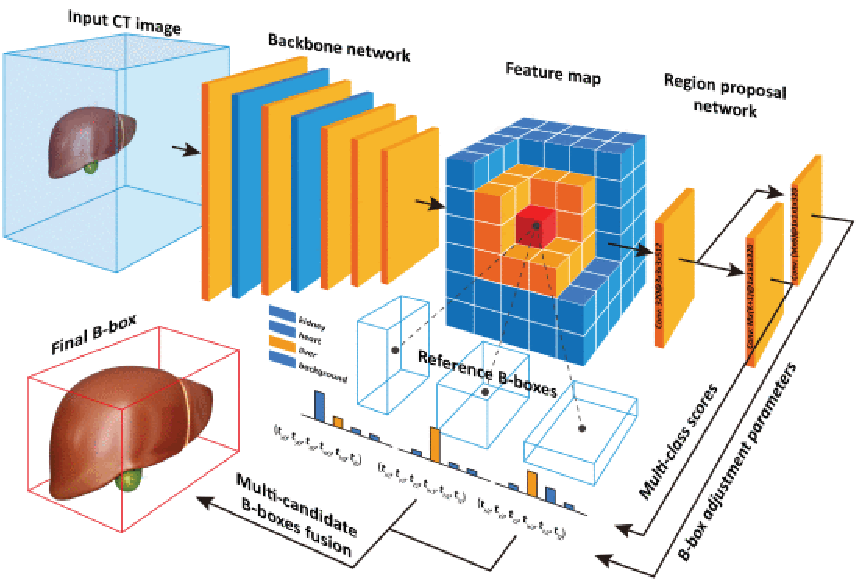
5 Author(s) Xuanang Xu ; Fugen Zhou ; Bo Liu ; Dongshan Fu ; Xiangzhi Bai **View All Authors**

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_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

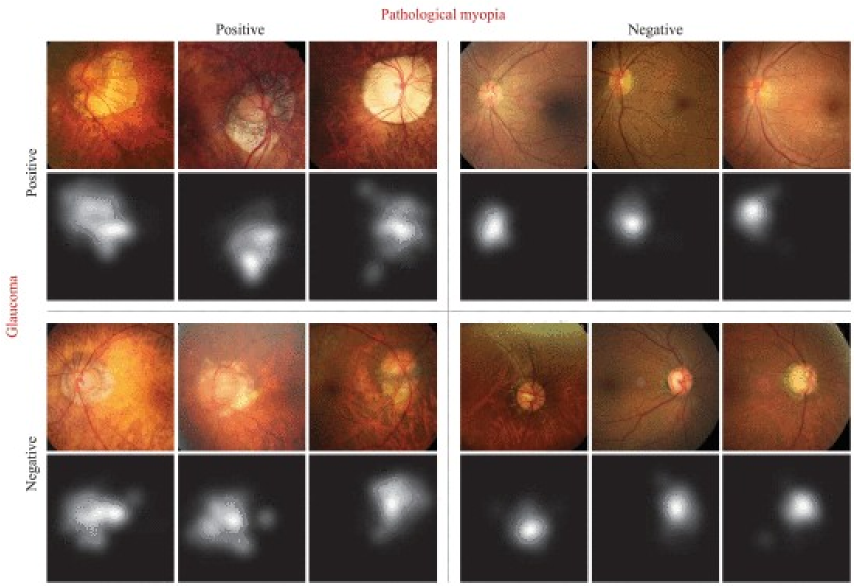
5 Author(s) Xuanang Xu ; Fugen Zhou ; Bo Liu ; Dongshan Fu ; Xiangzhi Bai **View All Authors**

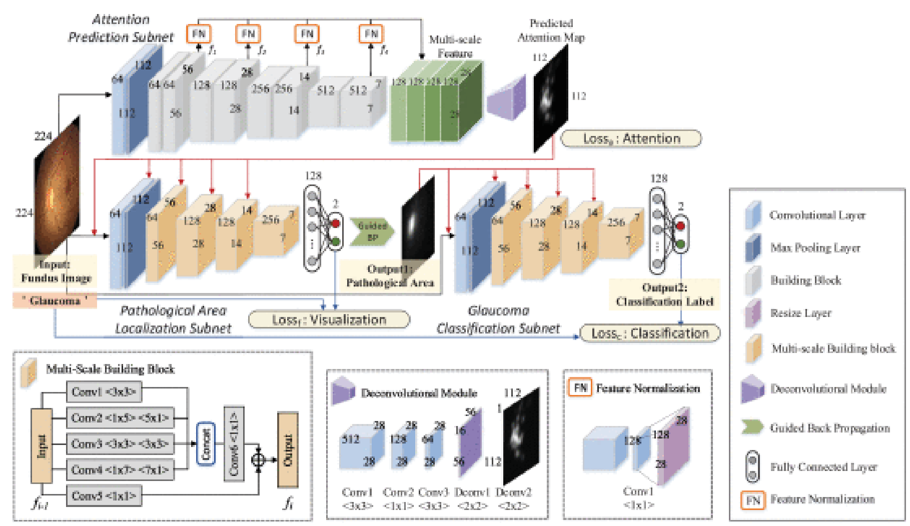
1 Paper Citation

1554 Full Text Views

Detection with Attention

- Detection of glaucoma
- Capturing attention obtained from ophthalmologists through a simulated eye-tracking experiment.
- Train the attentation module together with the classification.





Journals & Magazines > IEEE Transactions on Medical ... > Volume: 39 Issue: 2

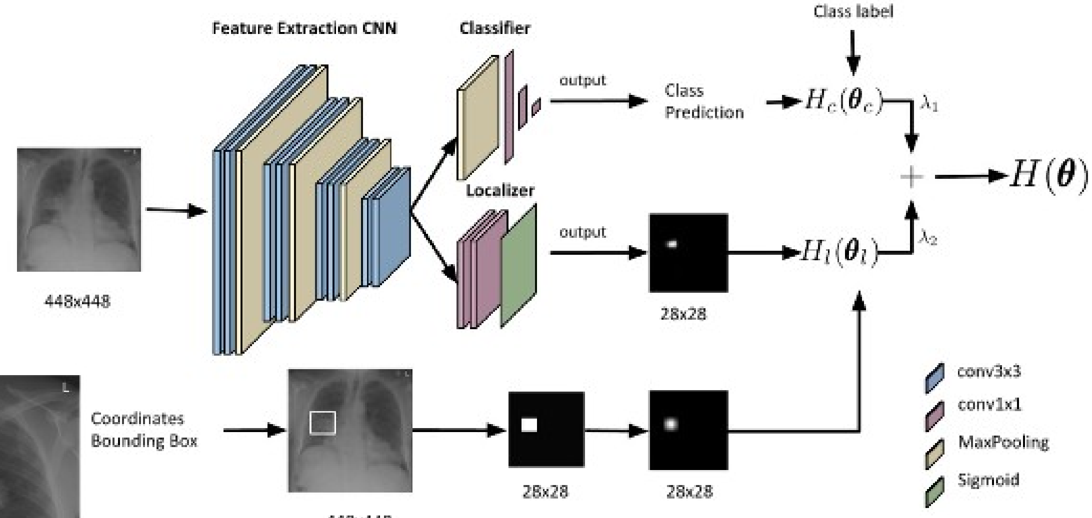
A Large-Scale Database and a CNN Model for Attention-Based Glaucoma Detection

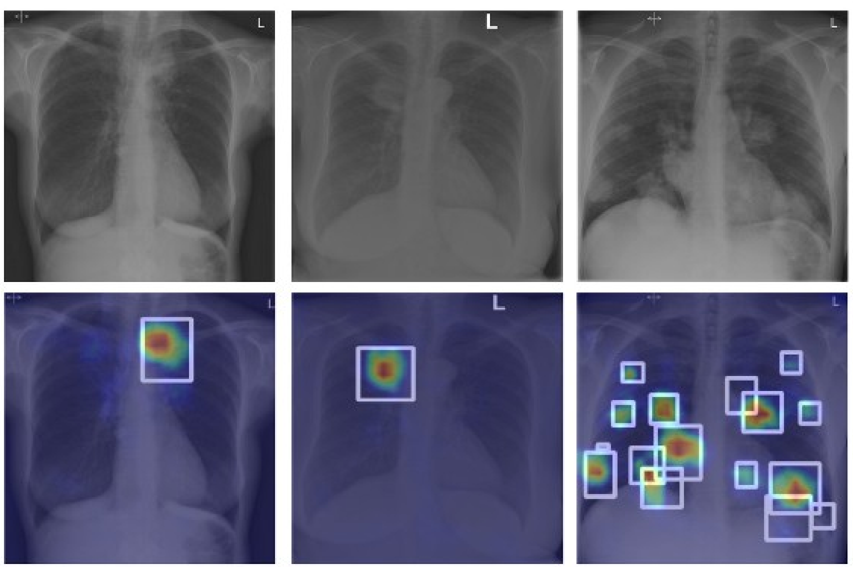
Publisher: IEEE

9 Author(s) Liu Li ; Mai Xu ; Hanruo Liu ; Yang Li ; Xiaofei Wang ; Lai Jiang ; Zulin Wang ; Xiang F... **View All Authors**

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

ELSEVIER

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,*]

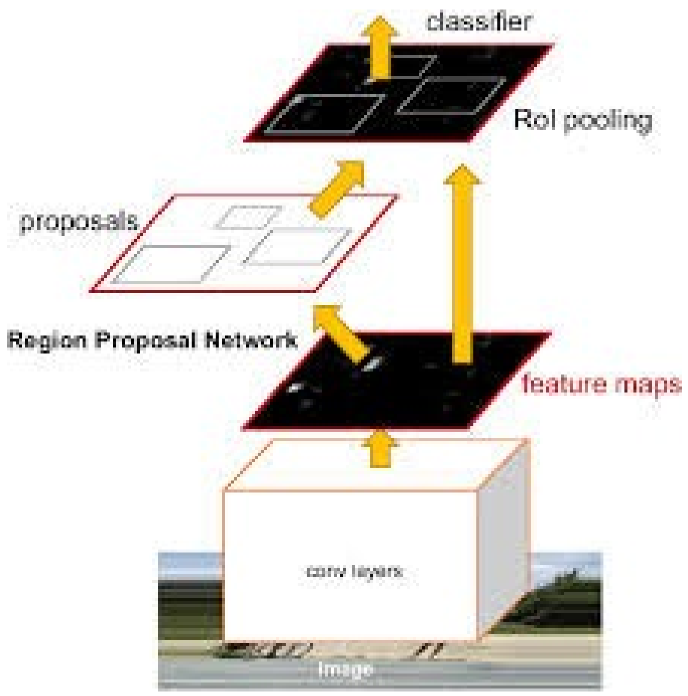
[a] Department of Biomedical Engineering, King's College London, London, UK
[b] Department of Radiology, Guy's & St Thomas' NHS Foundation Trust, London, UK
[c] Department of Medicine, Imperial College Healthcare NHS Trust, London, UK
[d] Department of Cancer Imaging, King's College London, London, UK

Conclusions

- 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, …

Lab Session!

- Detecting lesions in mammograms
  - Faster R-CNN



TorchVision

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

Course website: http://www.aramislab.fr/teaching/DLMI-2019-2020/
Piazza (for registered students): https://piazza.com/centralesupelec/spring2020/mvadlmi/