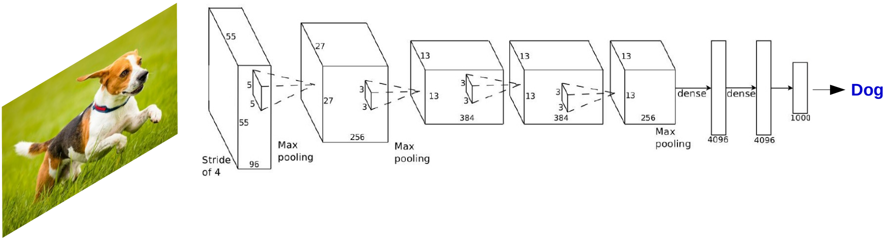
Region Based CNNs: Application to Object Recognition

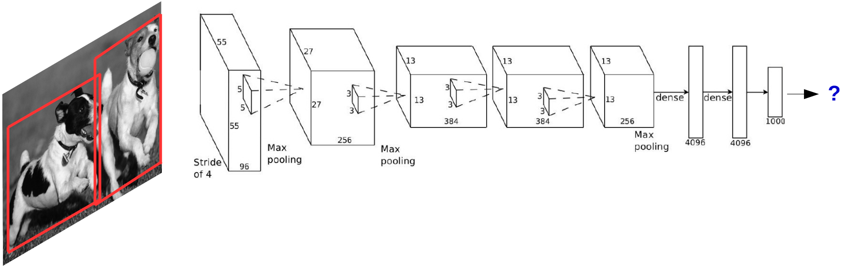
Alvaro Soto

Computer Science Department (DCC), PUC

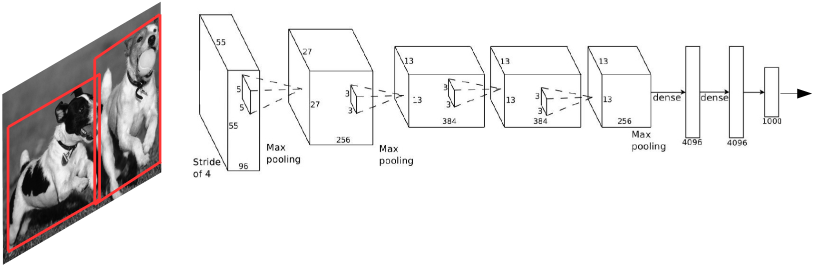
Object Recognition with CNNs



What about now?

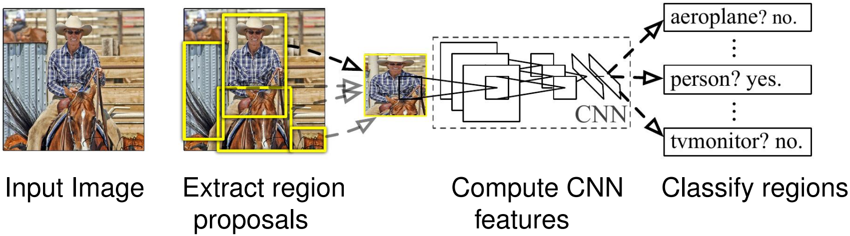


What about now?



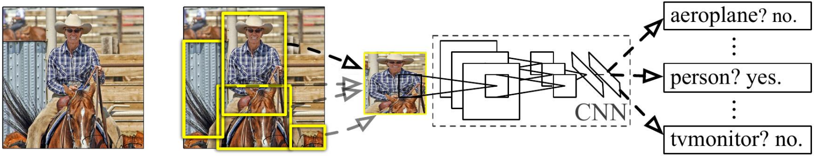
Dog ?
Dogs ?
Tennis ball ?
Park ?

Region based CNNs to the rescue: R-CNNs
R. Girshick et al., 2014



- How to extract the region proposal?
- How to manage region size?
- How can I train classifier?

Region based CNNs to the rescue: R-CNNs
R. Girshick et al., 2014



Input Image | Extract region proposals | Compute CNN features | Classify regions

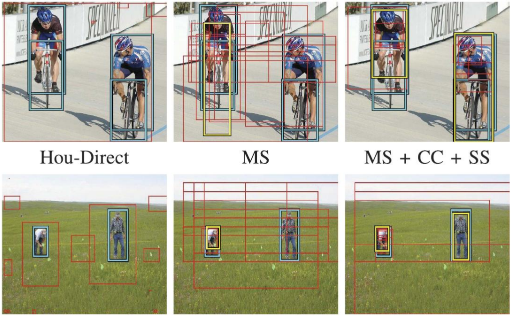
- How to extract the region proposal?
  - Use an saliency or objectness technique.

Objectness technique

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 34, NO. 11, NOVEMBER 2012 2189

Measuring the Objectness of Image Windows

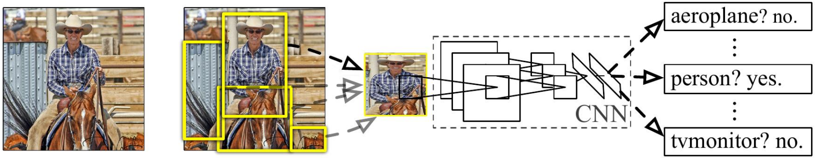
Bogdan Alexe, *Student Member*, *IEEE*, Thomas Deselaers, *Member*, *IEEE*, and Vittorio Ferrari, *Member*, *IEEE*



There are several alternative techniques:

- Selective Search [van de Sande, Uijlings et al.] (Used in this work)
- Objectness [Alexe et al.]
- Category independent object proposals [Endres & Hoiem]
- CPMC [Carreira & Sminchisescu]

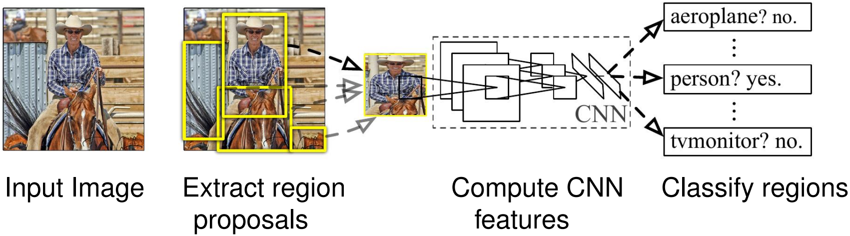
Region based CNNs to the rescue: R-CNNs
R. Girshick et al., 2014



Input Image | Extract region proposals | Compute CNN features | Classify regions

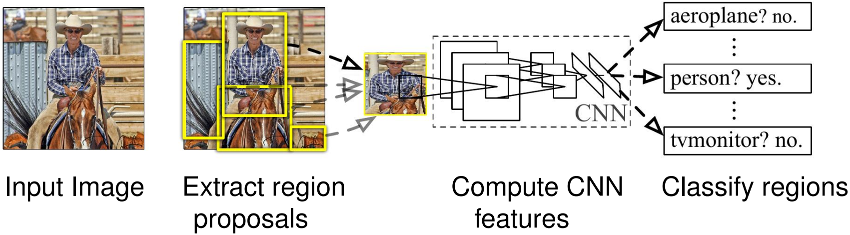
- How to extract the region proposal?
  - Use an saliency or objectness technique
  - This work selects around 2K region proposals for each image.

Region based CNNs to the rescue: R-CNNs
R. Girshick et al., 2014



- How to manage region size?
  - Scale all regions to size used by CNN (227x277 pixels).

Region based CNNs to the rescue: R-CNNs
R. Girshick et al., 2014



- How can I train classifier?
  - Use pre-trained CNN to extract features.
  - Use this features to train a classifier, ex. an SVM in this work.
  - As an alternative, it is possible to adapt head of the CNN to target classes.

Pascal dataset

Visual Object Classes Challenge 2012 (VOC2012)

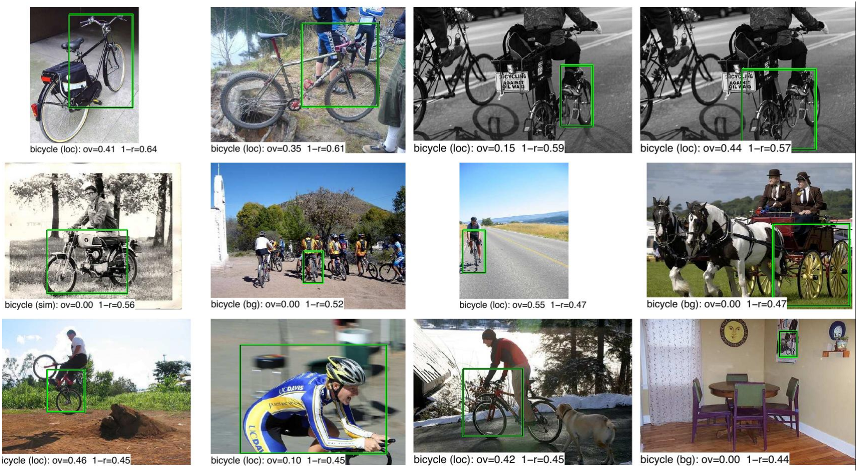
PASCAL2
Pattern Analysis, Statistical Modelling and Computational Learning

http://host.robots.ox.ac.uk/pascal/VOC/

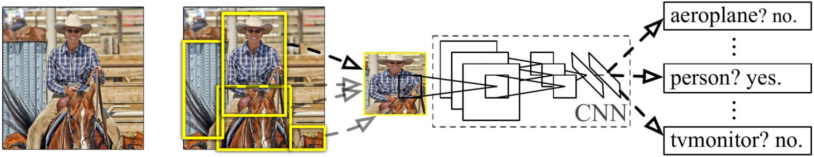
Results on Pascal dataset

	VOC 2007	VOC 2010
DPM v5 (Girshick et al. 2011)	33.7%	29.6%
UVA sel. search (Uijlings et al. 2013)		35.1%
Regionlets (Wang et al. 2013)	41.7%	39.7%
SegDPM (Fidler et al. 2013)		40.4%
R-CNN	54.2%	50.2%
R-CNN + bbox regression	**58.5%**	**53.7%**

Results on Pascal dataset



Region based CNNs to the rescue: R-CNNs
R. Girshick et al., 2014



Input Image | Extract region proposals | Compute CNN features | Classify regions

Any problem?

Fast R-CNN
Ross Girshick, 2015

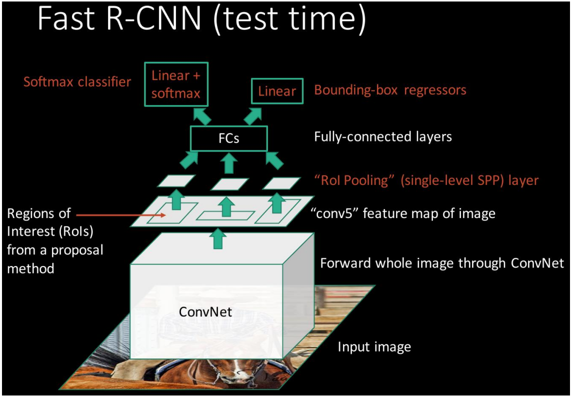


Image credit Ross Girshick.

Problem 1: Slow at test-time due to independent forward passes of the CNN.

Solution 1: Share convolutional layers between proposals, How?.

Fast R-CNN
Ross Girshick, 2015

Region of interest (RoI) alignment

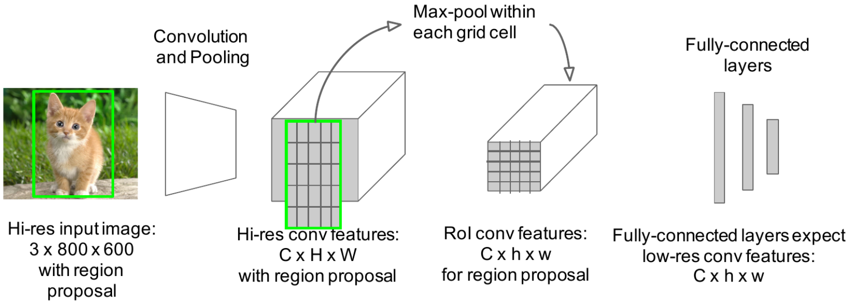
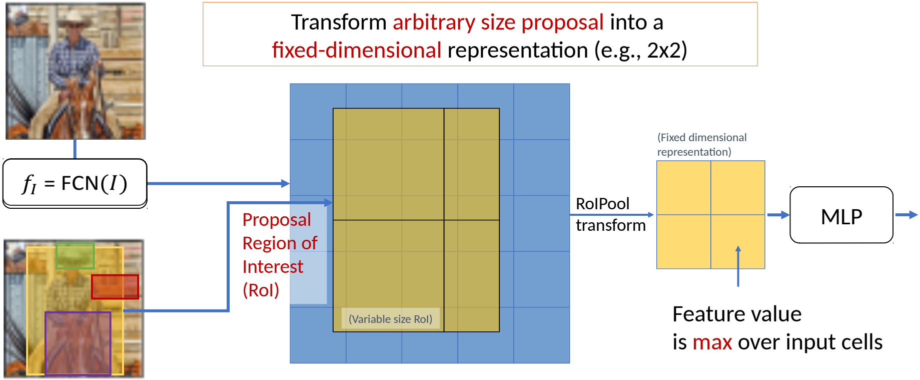


Image credit Justin Johnson .

Region of interest (RoI) alignment

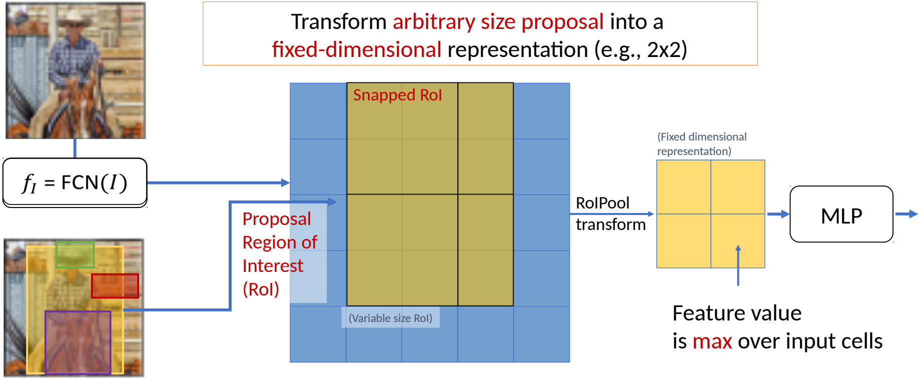
Transform arbitrary size proposal into a fixed-dimensional representation (e.g., 2x2)



Fast R-CNN
Ross Girshick, 2015

Region of interest (RoI) alignment

Transform arbitrary size proposal into a fixed-dimensional representation (e.g., 2x2)



Fast R-CNN
Ross Girshick, 2015

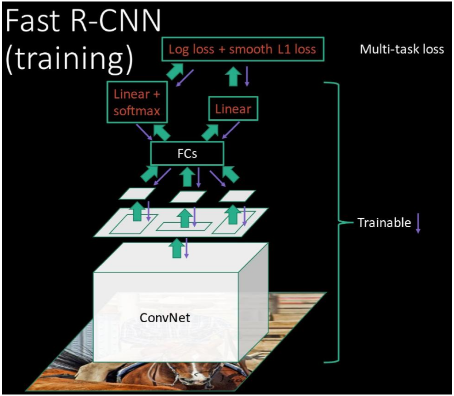


Image credit Ross Girshick.

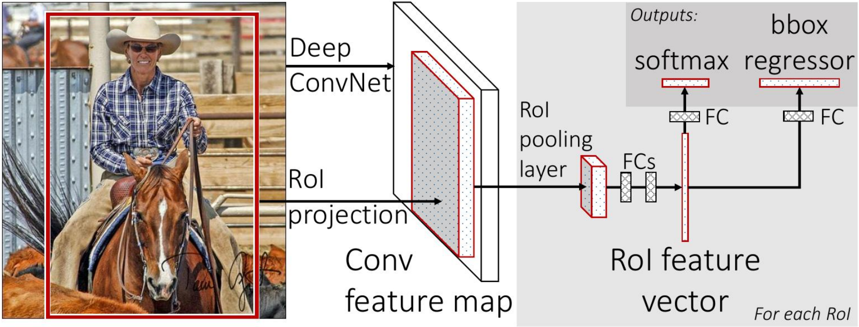
Fast R-CNN
Ross Girshick, 2015

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

Fast R-CNN
Ross Girshick, 2015

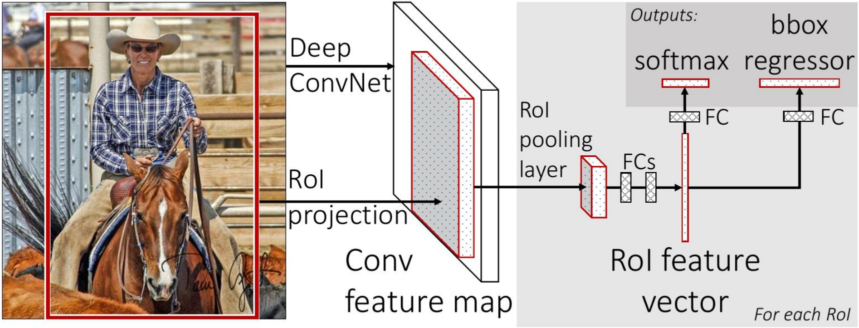
	R-CNN	**Fast R-CNN**
Test time per image	47 seconds	**0.32 seconds**
(Speedup)	1x	**146x**
Test time per image with Selective Search	50 seconds	**2 seconds**
(Speedup)	1x	**25x**

Fast R-CNN
Ross Girshick, 2015



Any problem?

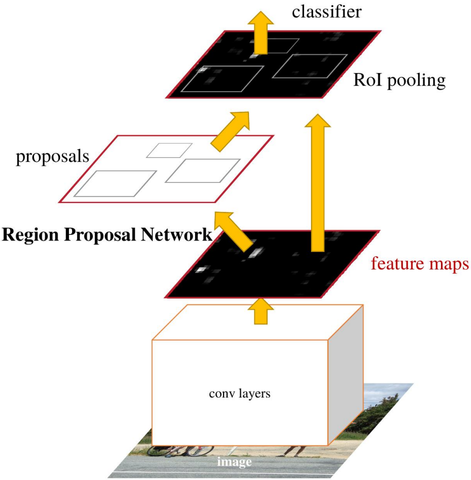
Fast R-CNN
Ross Girshick, 2015



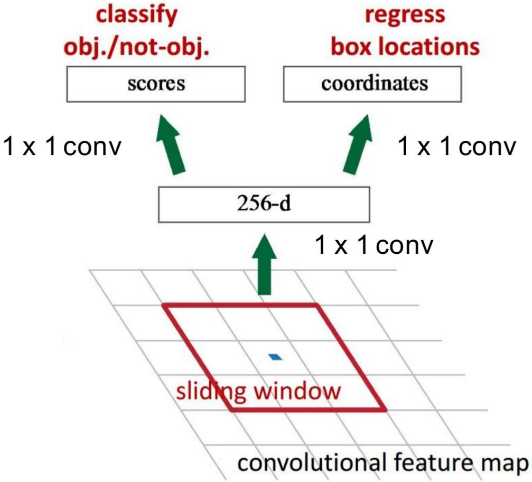
Problem: Region proposals are still given by an external method.

Solution: Make the CNN do region proposals too.

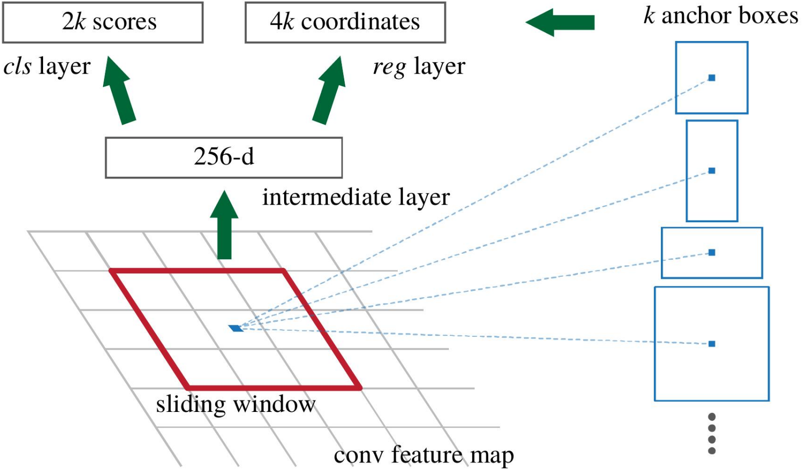
Faster R-CNN
Ren et al., 2015



Faster R-CNN
Ren et al., 2015

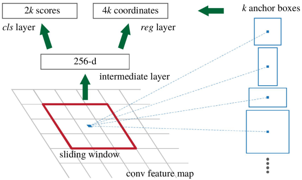


Faster R-CNN



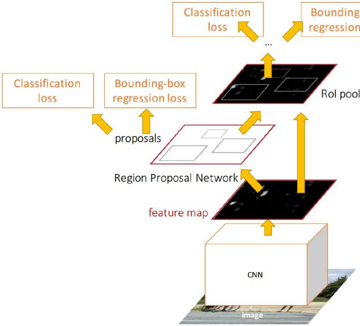
$2k$ scores
$4k$ coordinates
k anchor boxes
cls layer
reg layer
256-d
intermediate layer
sliding window
conv feature map

Faster R-CNN
Ren et al., 2015



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

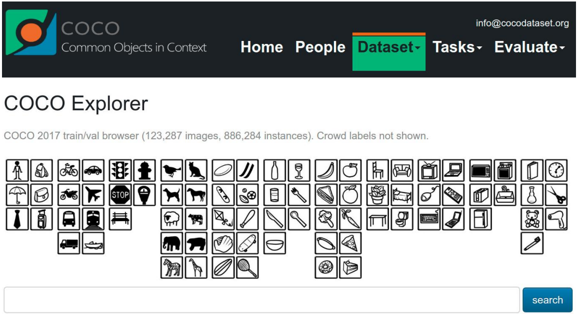
Faster R-CNN
Ren et al., 2015



Joint training, one network and four losses:

- RPN classification (anchor good / bad)
- RPN regression (anchor -> proposal)
- Fast R-CNN classification (over classes)
- Fast R-CNN regression (proposal -> box)

Microsoft-Coco dataset



https://cocodataset.org

Fast vs Faster R-CNN

model	system	conv	proposal	region-wise	total	rate
VGG	SS + Fast R-CNN	146	1510	174	1830	0.5 fps
VGG	RPN + Fast R-CNN	141	**10**	47	**198**	**5 fps**
ZF	RPN + Fast R-CNN	31	**3**	25	**59**	**17 fps**

method	proposals	training data	COCO val		COCO test-dev	
			mAP@.5	mAP@[.5, .95]	mAP@.5	mAP@[.5, .95]
Fast R-CNN [2]	SS, 2000	COCO train	-	-	35.9	19.7
Fast R-CNN [impl. in this paper]	SS, 2000	COCO train	38.6	18.9	39.3	19.3
Faster R-CNN	RPN, 300	COCO train	41.5	21.2	42.1	21.5
Faster R-CNN	RPN, 300	COCO trainval	-	-	**42.7**	**21.9**

Faster R-CNN
Ren et al., 2015

training data	COCO train		COCO trainval	
test data	COCO val		COCO test-dev	
mAP	@.5	@[.5, .95]	@.5	@[.5, .95]
baseline Faster R-CNN (VGG-16)	41.5	21.2		
baseline Faster R-CNN (ResNet-101)	48.4	27.2		
+box refinement	49.9	29.9		
+context	51.1	30.0	53.3	32.2
+multi-scale testing	53.8	32.5	**55.7**	**34.9**
ensemble			**59.0**	**37.4**