Deep Learning for Object detection & localization

R-CNN, Fast R-CNN, Faster R-CNN, YOLO, GAP, CAM, MSROI

Aaditya Prakash

March 15, 2017

Image classification

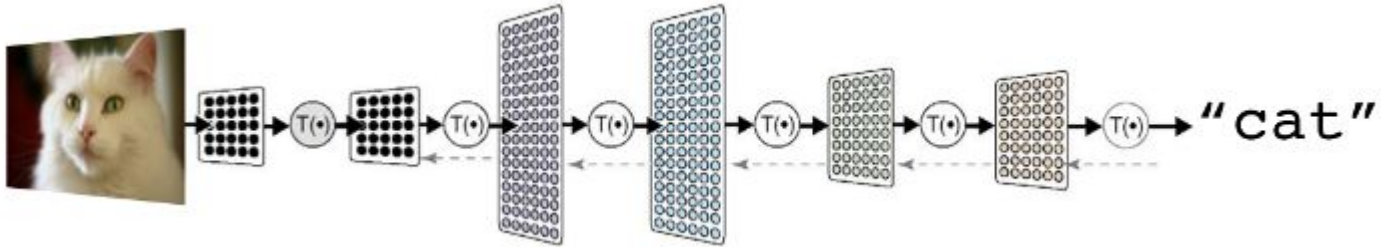
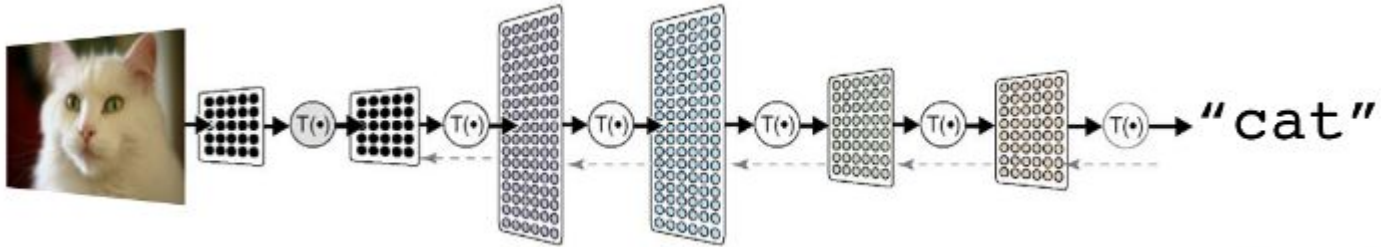
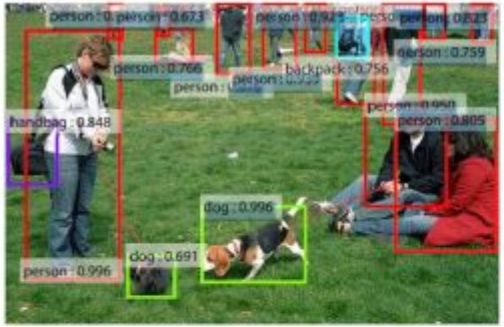


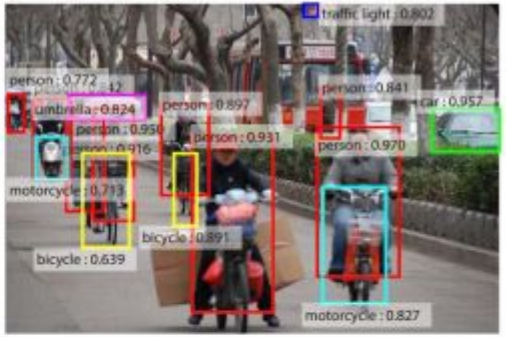
Image classification



- Whole of image is classified as one class.
- Every pixel is used to determine the class.
- Does not know/care if there is another object in the image.
- No separation of background/foreground, hence ‘tank-problem’.

Localization and Detection





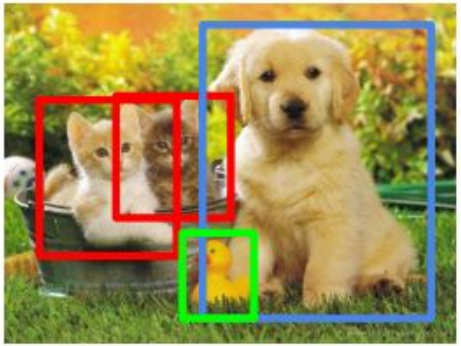
Source: cs231n

Classification vs Localization vs Detection vs Segmentation

CAT

CAT

Single object



CAT, DOG, DUCK

CAT, DOG, DUCK

Multiple objects

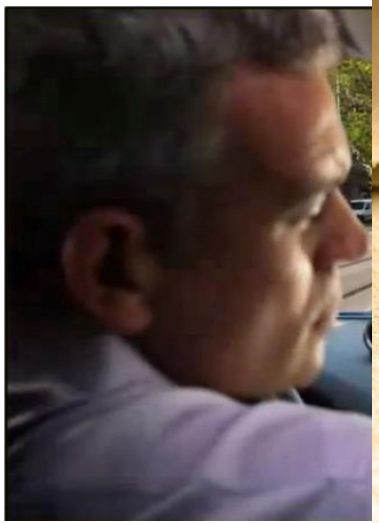
Source: cs231n

Detection - Fast and Accurate

MobilEye (video)

Nikon S60

Detection - Fast and Accurate



MobilEye (video)

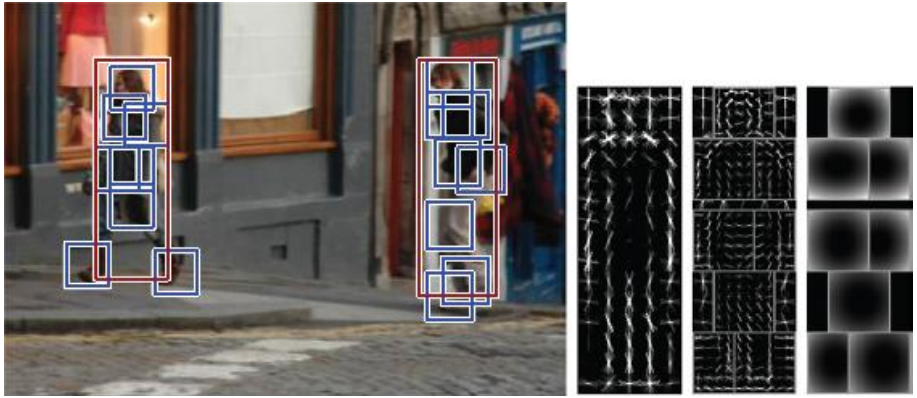
Source: Ross Girshick

Demo

Source: YOLO

Before deep learning

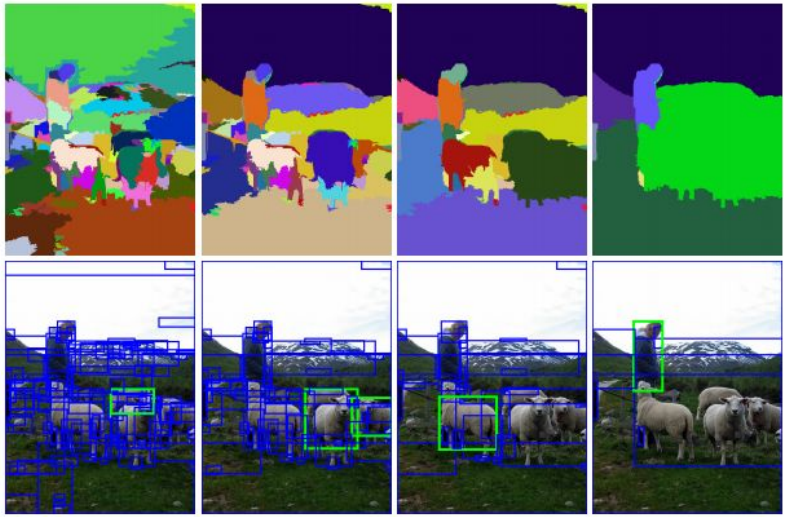
- Deformable Parts Model
  - low-level features based on HOG, SVM classifier
  - mAP: 33%



Source: Pedro Felzenszwalb

Before deep learning

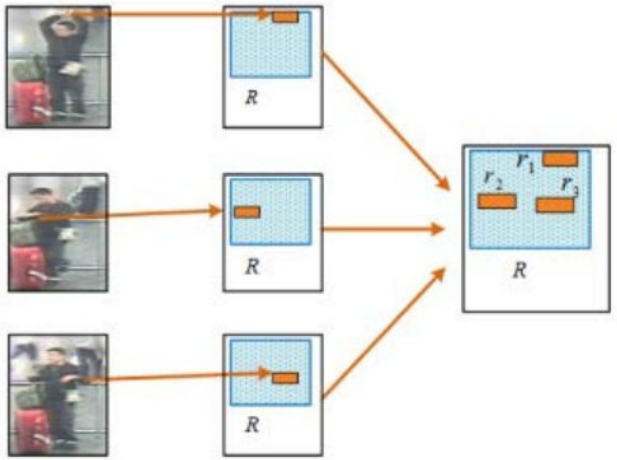
- Deformable Parts Model
  - low-level features based on HOG, SVM classifier
  - mAP: 33%
- Selective Search
  - Bag-of-words with SIFT features, merge similar segments on segmentation
  - mAP: 35%



Source: Uijlings et al

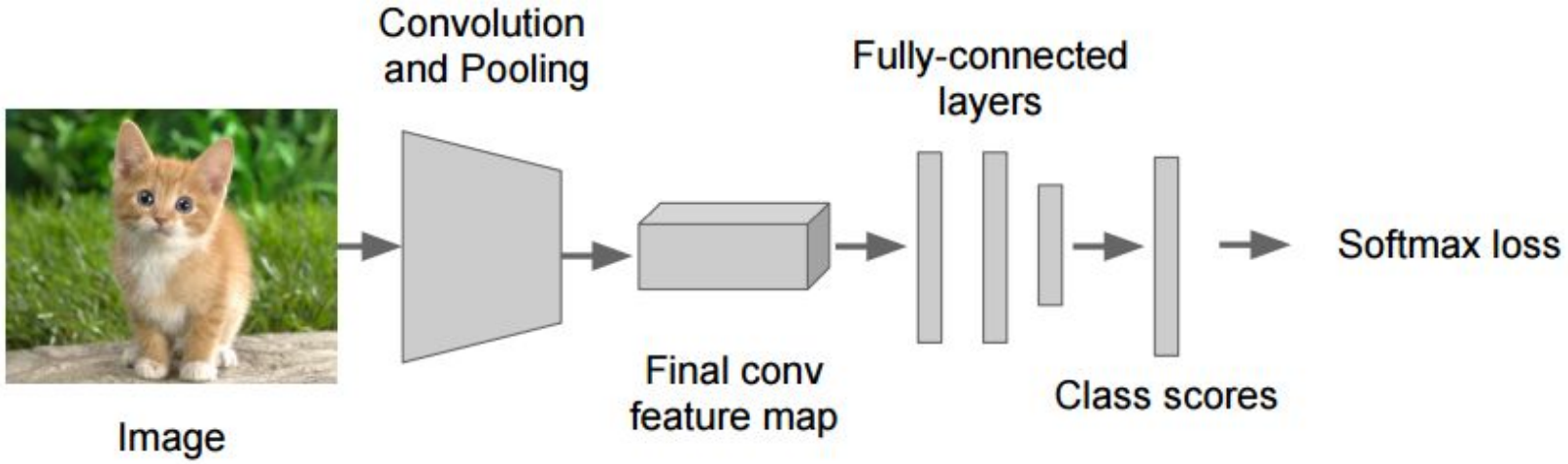
Before deep learning

- Deformable Parts Model
  - low-level features based on HOG, SVM classifier
  - mAP: 33%
- Selective Search
  - Bag-of-words with SIFT features, merge similar segments found by segmentation
  - mAP: 35%
- Regionlets
  - HOG, LBP, Covariance features, relative position of sub-parts
  - mAP: 40%



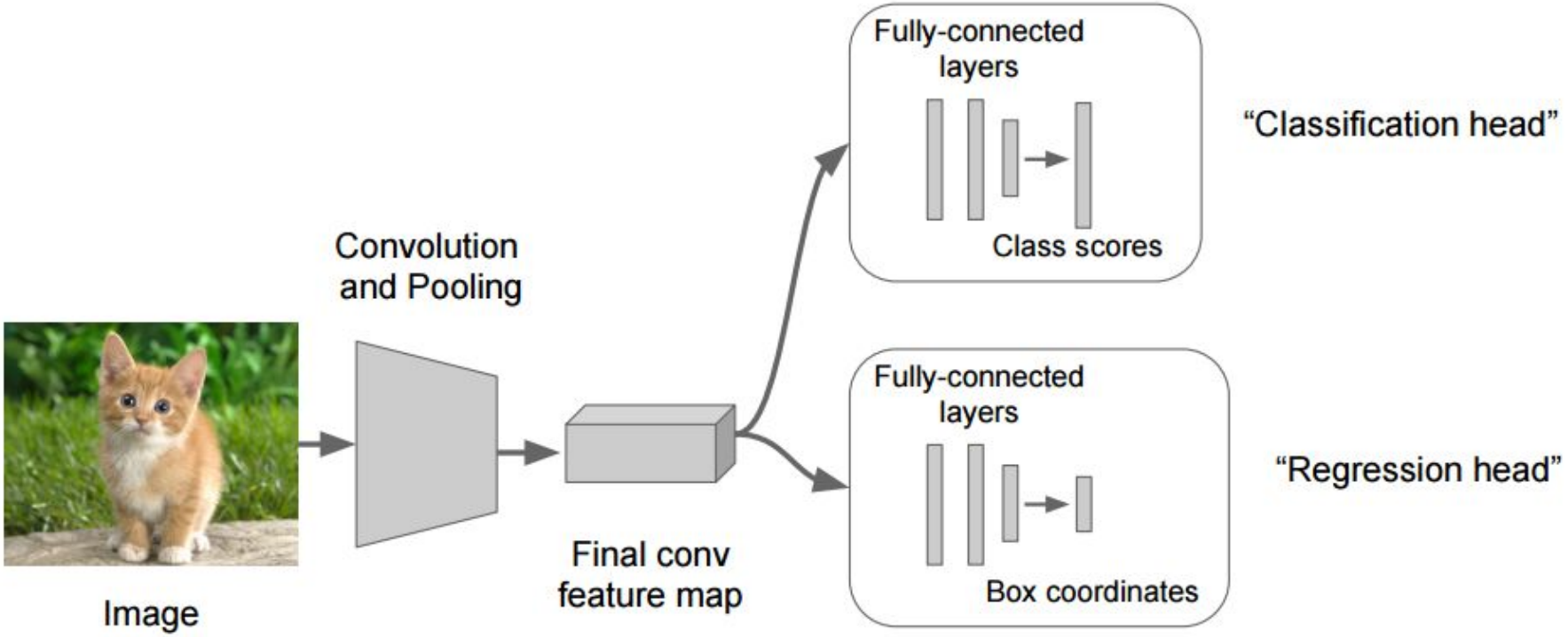
Source: Wang et al

Naive way to use CNN for localization - Regression



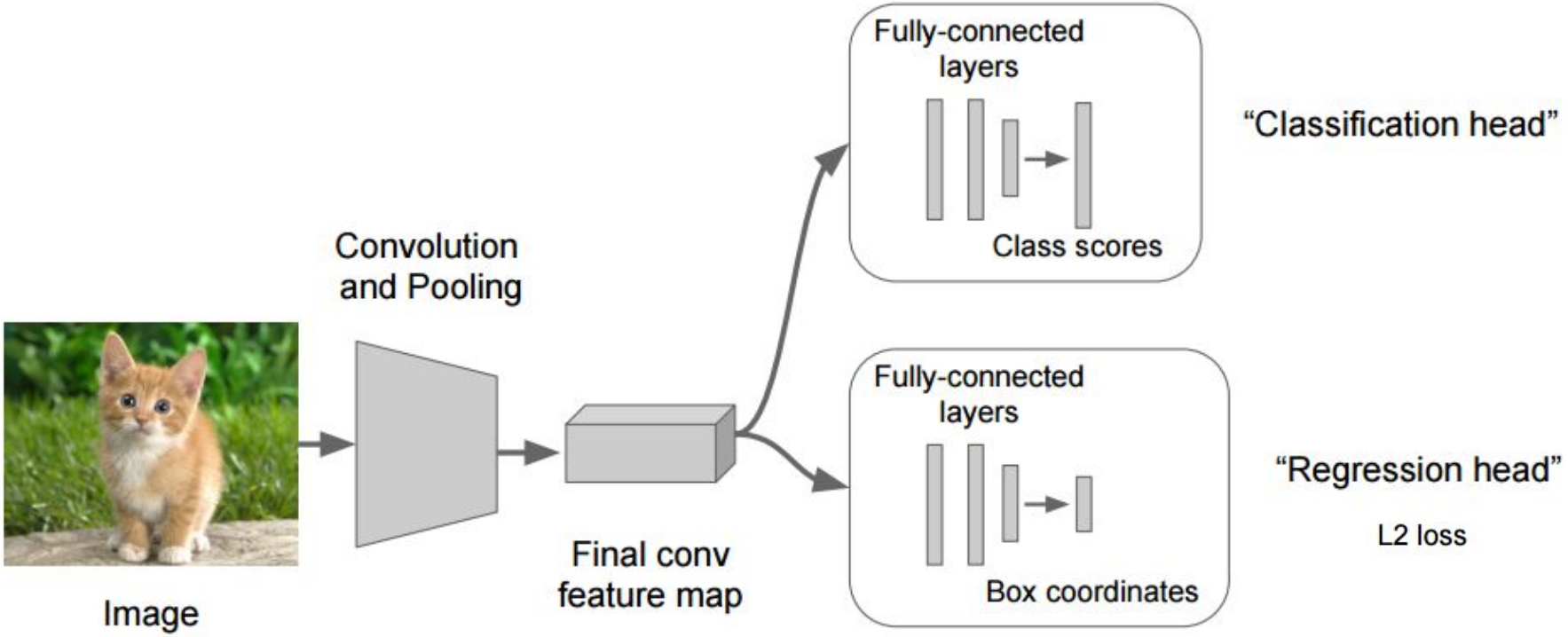
Source: cs231n

Naive way to use CNN for localization - Regression



Source: cs231n

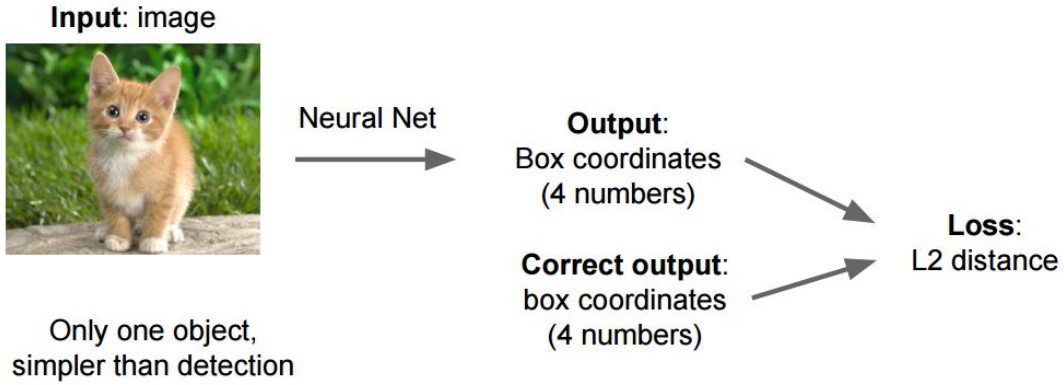
Naive way to use CNN for localization - Regression



Source: cs231n

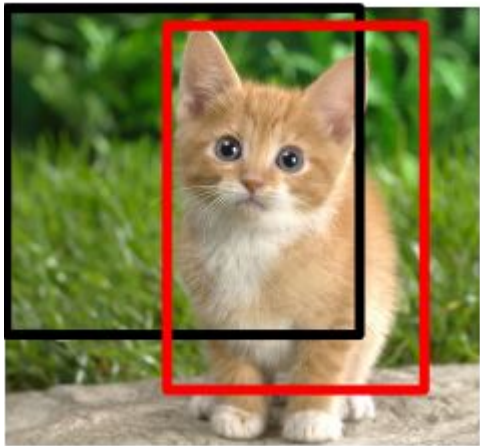
Naive way to use CNN for localization - Regression

- Limited to ‘one’ class per image
- Limited to ‘one’ instance of object per image
- Cannot separate object parts - like ear, head, eyes
- Limited to ‘rectangle’ object boundaries



Source: cs231n

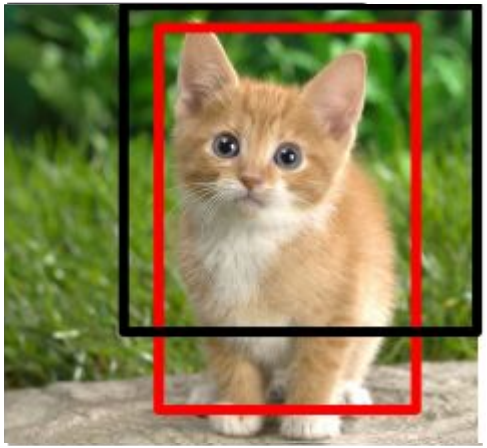
Better way - Sliding window



0.5	

Source: cs231n

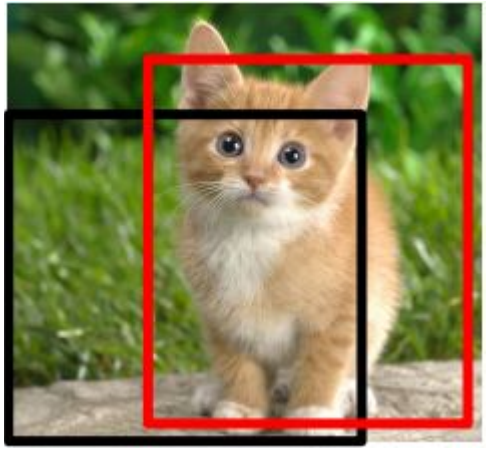
Better way - Sliding window



0.5	0.75

Source: cs231n

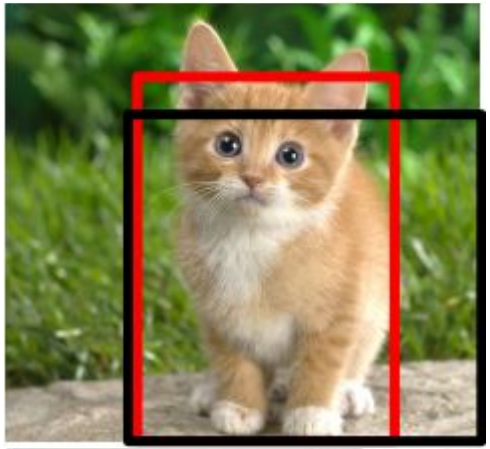
Better way - Sliding window



0.5	0.75
0.6	

Source: cs231n

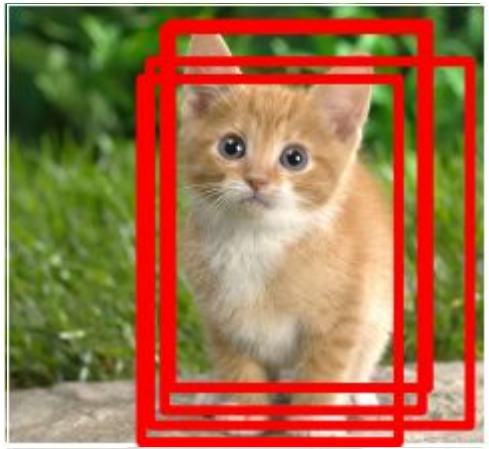
Better way - Sliding window



0.5	0.75
0.6	0.8

Source: cs231n

Better way - Sliding window



0.5	0.75
0.6	**0.8**

Source: cs231n

Imagenet results (Localization)

AlexNet: Localization method not published

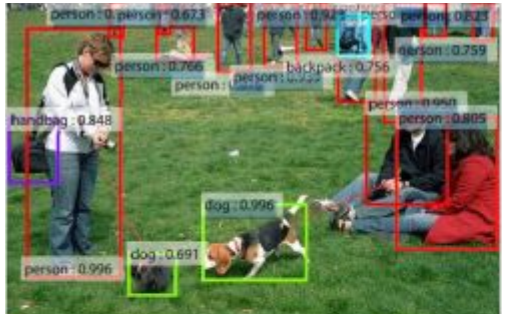
Overfeat: Multiscale convolutional regression with box merging

VGG: Same as Overfeat, but fewer scales and locations; simpler method, gains all due to deeper features

ResNet: Different localization method (RPN) and much deeper features

Can we do detection as regression?

- Four corners for every object, many numbers to regress.
- No idea on how many objects to expect.



Can we do detection as classification?

- How many boxes should we check? All of them?
- At what scale? Some dogs are smaller (or far away)

Can we do detection as regression?

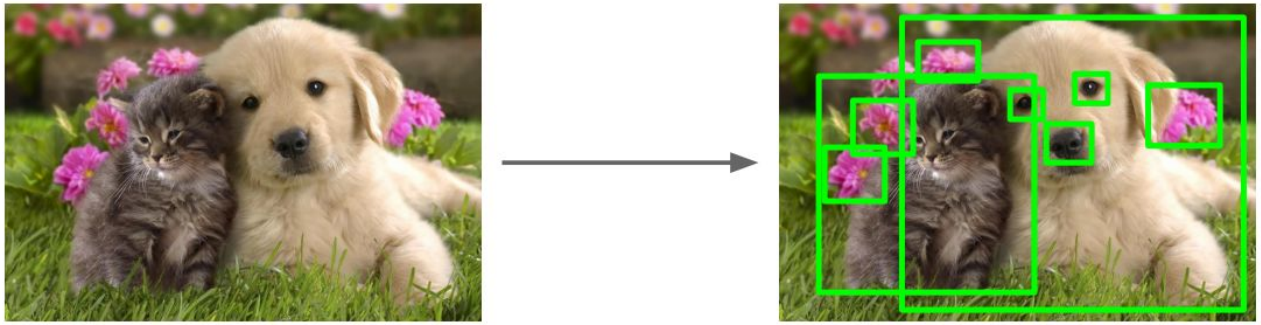
- Four corners for every object, many numbers to regress.
- No idea on how many objects to expect.

Solution: Only only at few places

Can we do detection as classification?

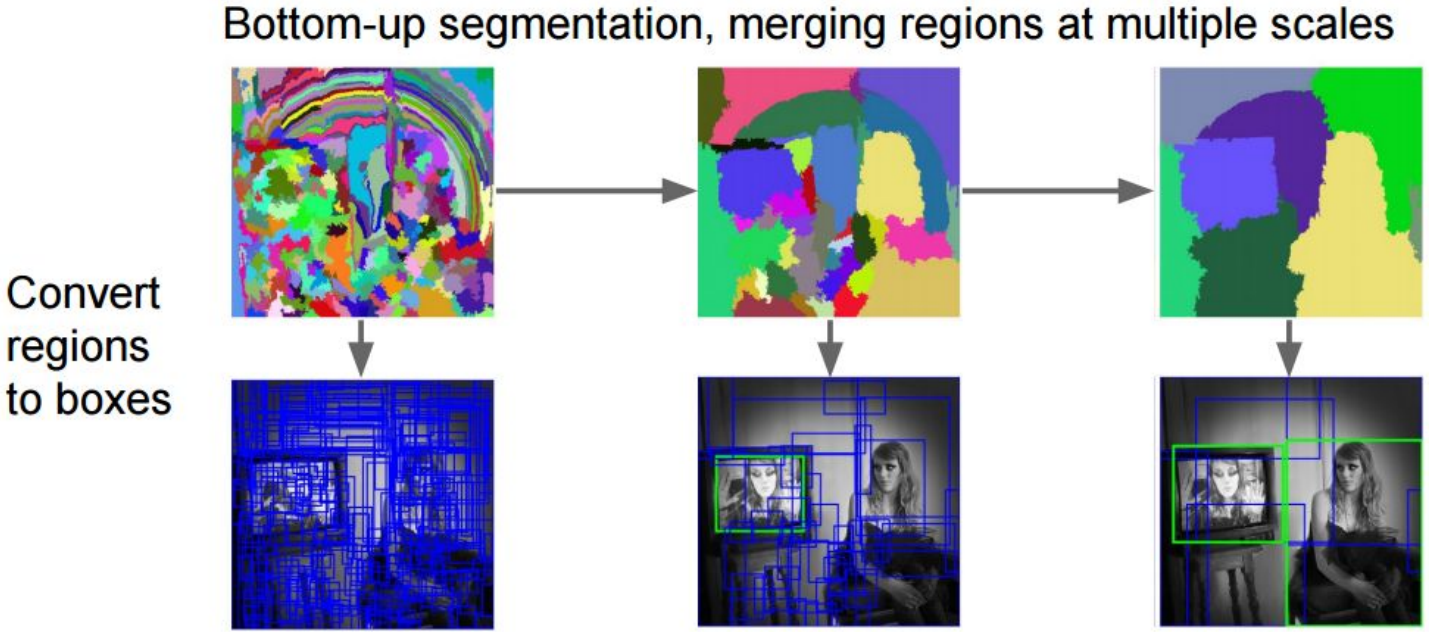
- How many boxes should we check? All of them?
- At what scale? Some dogs are smaller (or far away)

Better solution - Region proposals

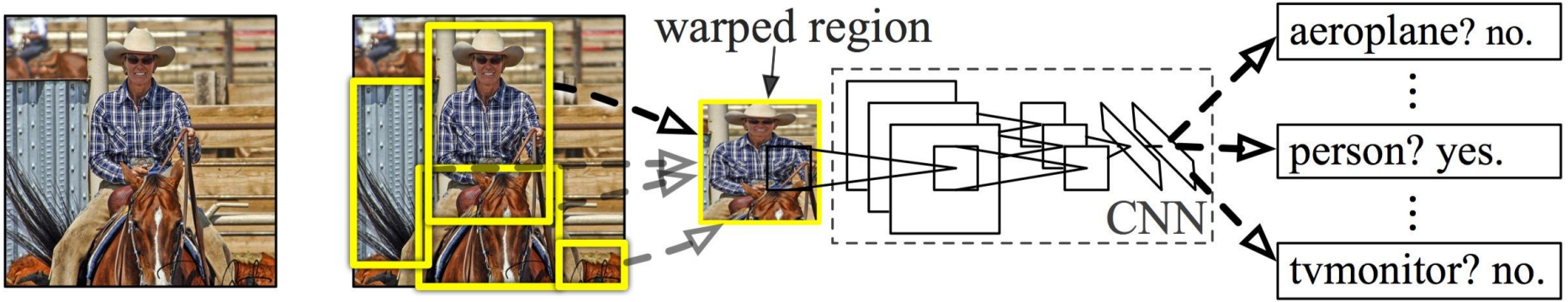


- Find ‘regions-of-interest’ that are likely to contain some objects
- A simple detector which just says there is something of interest and not necessarily about the class
- These region proposals could be just blob of pixels or ‘super-pixels’.
- There are lot of techniques, like BING, CPMC, EdgeBoxes, Geodesic, MCG etc

Region Proposals - Selective Search



Region based CNN (R-CNN)



R-CNN Training

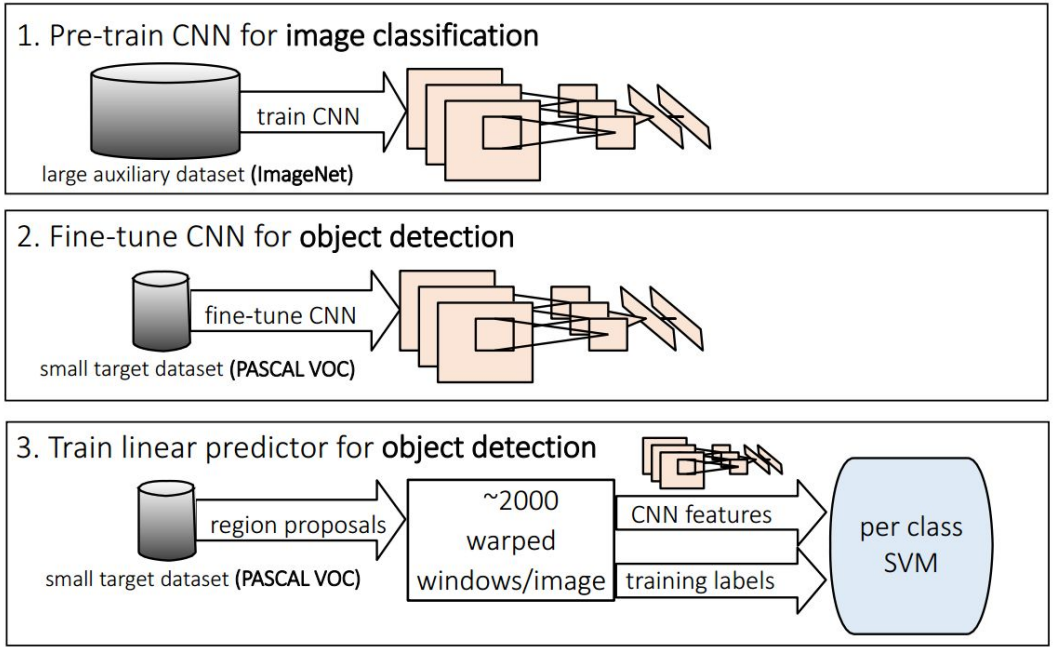
1. [offline] ~~M~~ Pre-train a ConvNet for ImageNet classification

2. ~~M'~~ Fine-tune M for object detection (softmax classifier + log loss) [add background class]

3. ~~F~~ Cache feature vectors to disk using M'

4. Train *post hoc* linear SVMs on F (hinge loss) [binary classifier]

5. Train *post hoc* linear bounding-box regressors on F (squared loss)



Source: E Mercan, R Girshick

R-CNN Results/Analysis

mAP - mean Average Precision
- Average of Average Precision for each class.
- True positive if IoU with ground-truth box > 0.5

1. Very slow (upto 20 seconds per image on K40 GPU).
2. Needs to run full CNN pass on each proposed region.
3. Not end-to-end training.
4. CNN features are not updated with results of training SVM
5. Better results than any other system

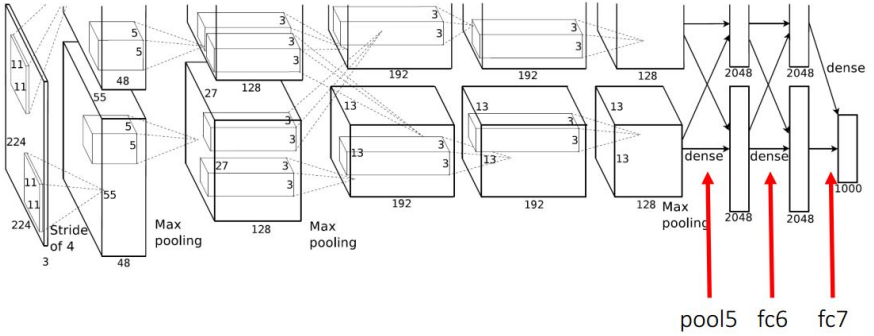
Source: E Mercan

R-CNN Results/Analysis

mAP - mean Average Precision
- Average of Average Precision for each class.
- True positive if IoU with ground-truth box > 0.5.
- AP is area under the Precision-Recall curve.

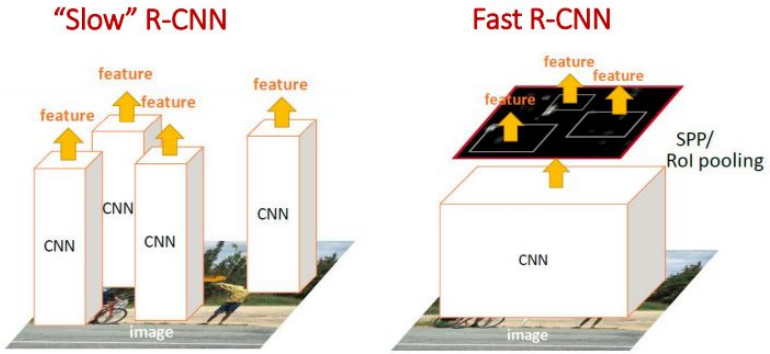
1. Very slow (upto 20 seconds per image on K40 GPU).
2. Needs to run full CNN pass on each proposed region.
3. Not end-to-end training.
4. CNN features are not updated with results of training SVM
5. Better results than any other system

		VOC 2007	VOC 2010
reference	DPM v5 (Girshick et al. 2011)	33.7%	29.6%
	UVA sel. search (Uijlings et al. 2012)		35.1%
	Regionlets (Wang et al. 2013)	41.7%	39.7%
pre-trained only	R-CNN $pool_5$	44.2%	
	R-CNN fc_6	46.2%	
	R-CNN fc_7	44.7%	
fine-tuned	R-CNN $pool_5$	47.3%	
	R-CNN fc_6	53.1%	
	R-CNN fc_7	54.2%	50.2%%
	R-CNN fc_7 (Bounding Box regression)	**58.5%**	53.7%



Source: E Mercan

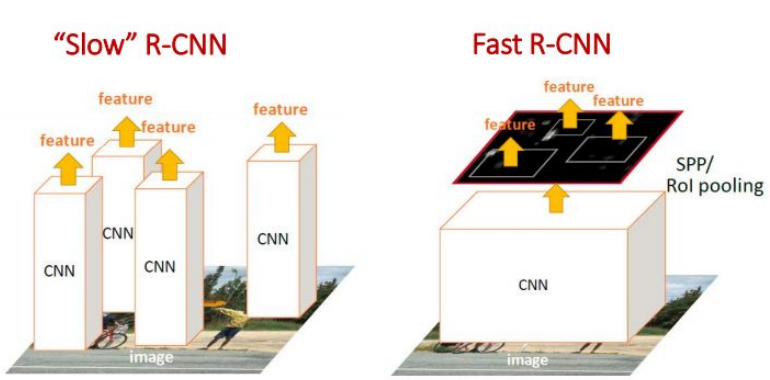
Fast R-CNN

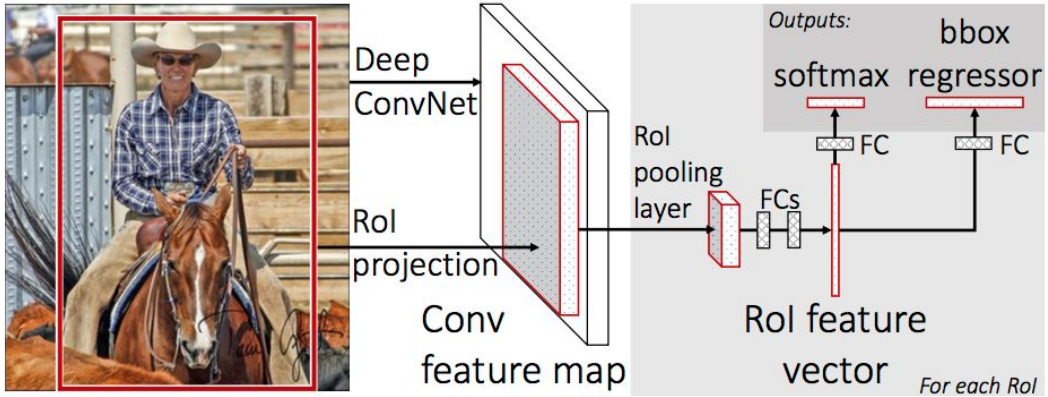


- Share computation of convolutional layers between 'regions of proposal'

Source: Ross Girshick

Fast R-CNN





- Share computation of convolutional layers between 'regions of proposal'
- Train the whole system end-to-end
  - no more post-hoc classifiers
  - No caching features to disk

Source: Ross Girshick, Kaiming He

Fast R-CNN

True box coordinates

Predicted box coordinates

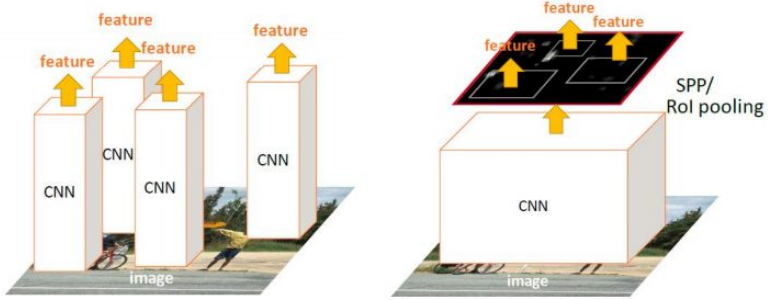
$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1]L_{\text{loc}}(t^u, v)$$

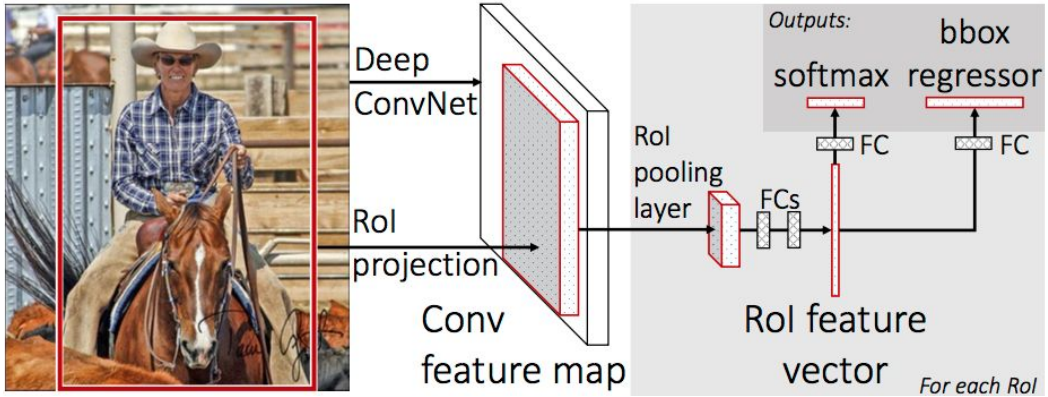
Predicted class scores

True class scores

Log loss

Smooth L1 loss



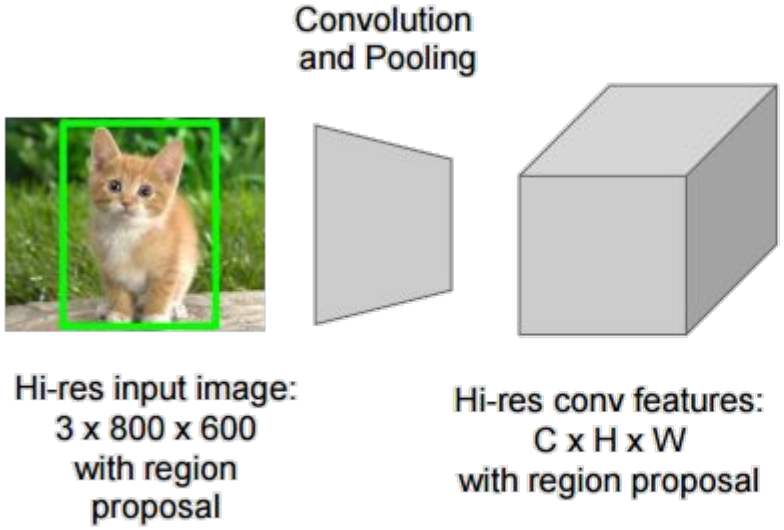


- Share computation of convolutional layers between 'regions of proposal'
- Train the whole system end-to-end
  - no more post-hoc classifiers
  - No caching features to disk

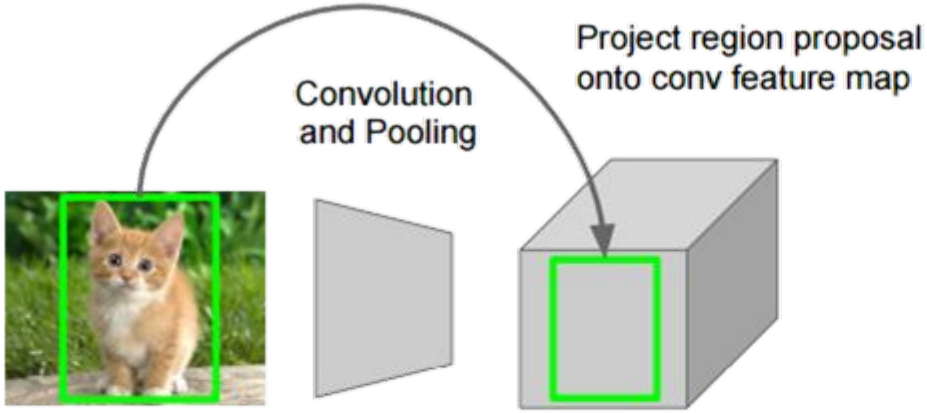
- Positive samples are those with IoU overlap with ground-truth bounding box > 0.5.
- Negative samples are in interval [0.1,0.5)
- 25%/75% ratio of Positive to Negative samples

Source: Ross Girshick, Kaiming He

Fast R-CNN: Region of Interest Pooling

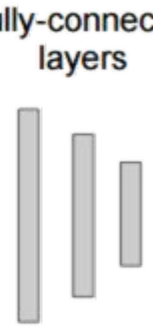


Fast R-CNN: Region of Interest Pooling



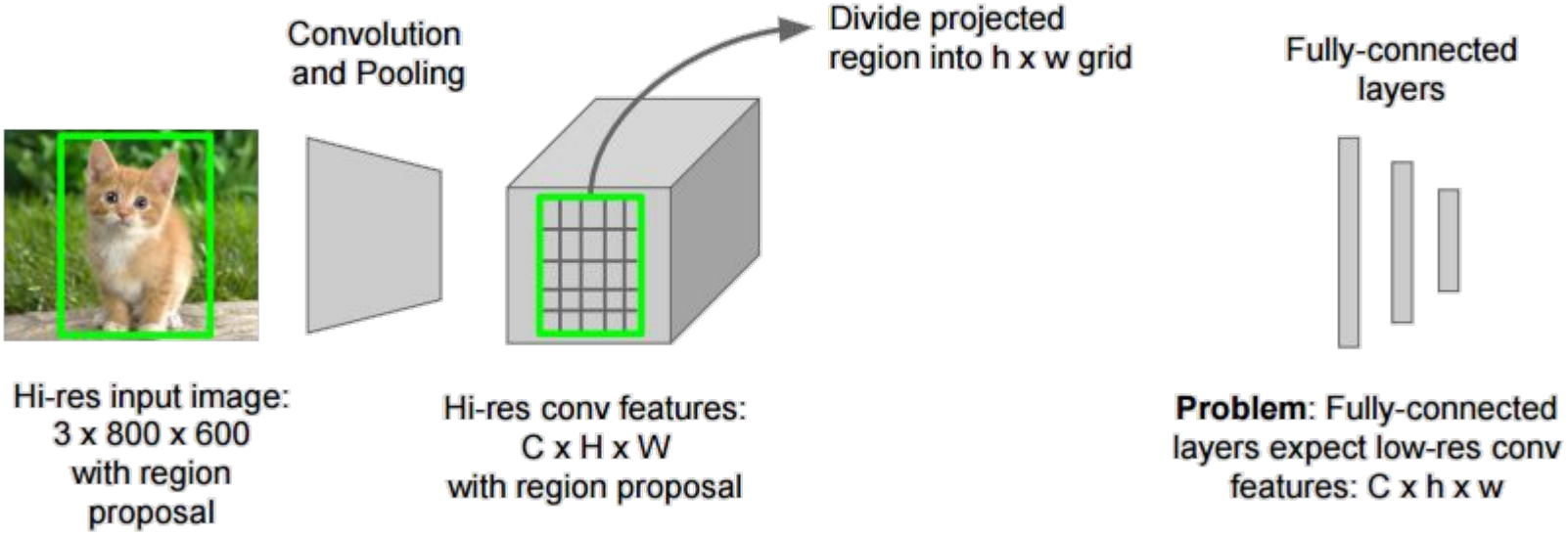
Hi-res input image:
3 x 800 x 600
with region
proposal

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

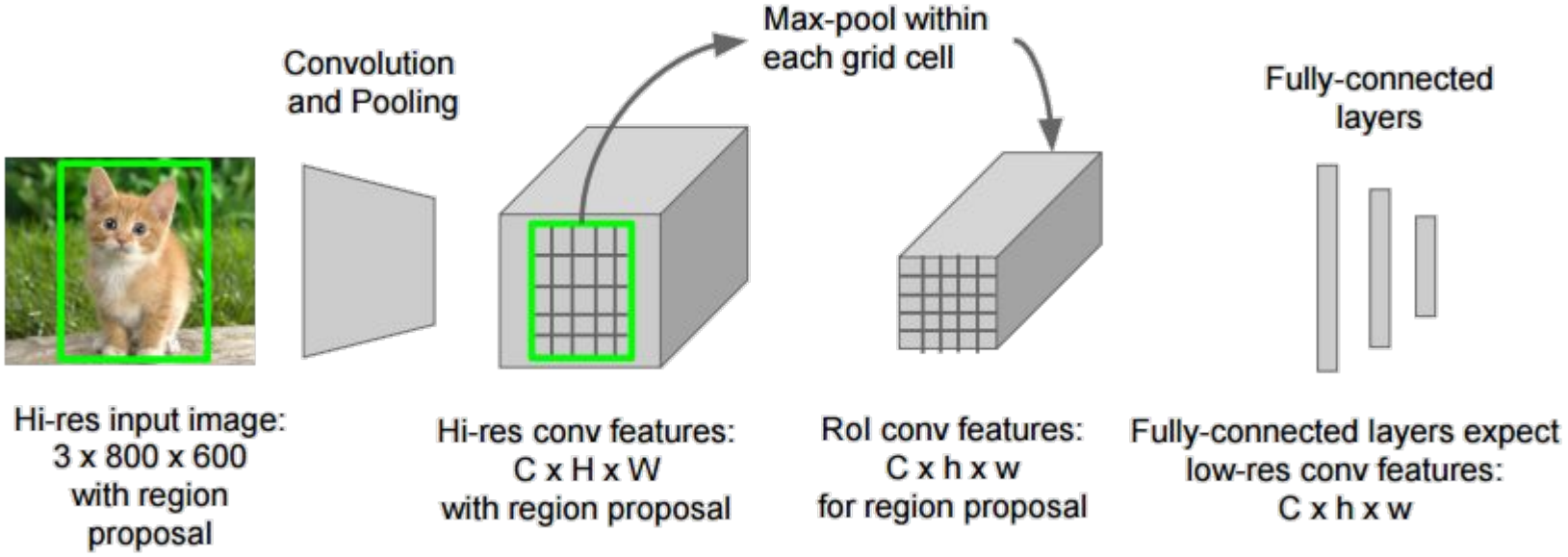


Problem: Fully-connected layers expect low-res conv features: C x h x w

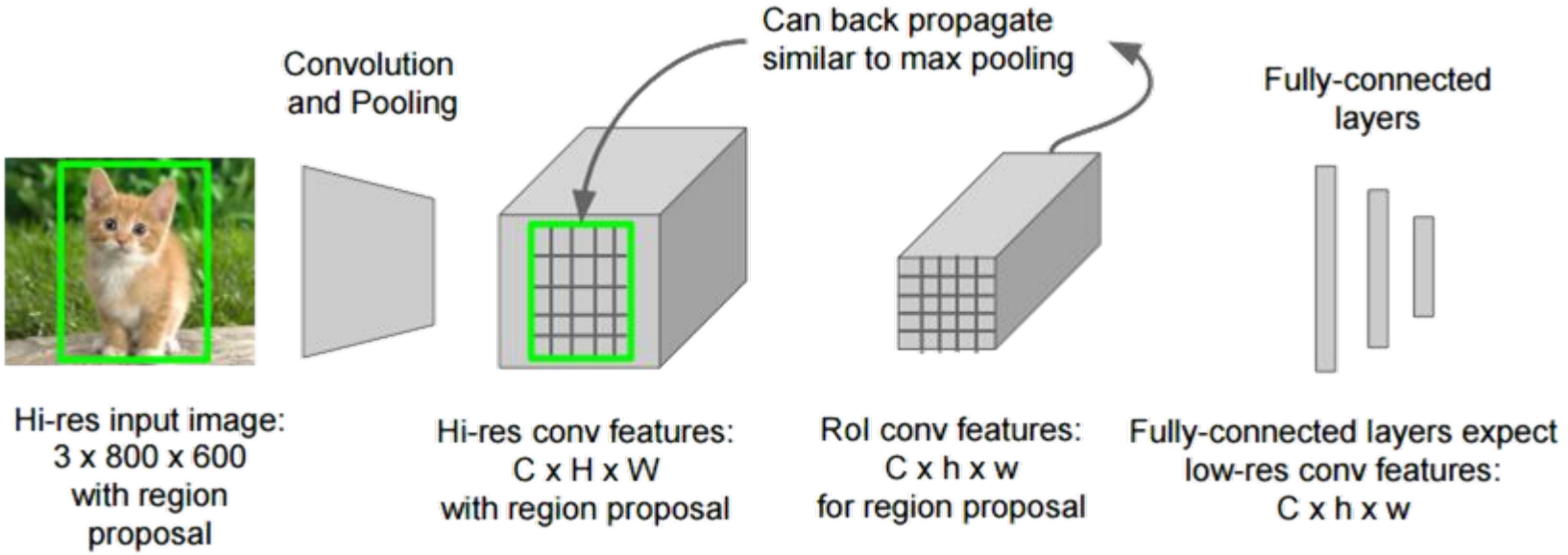
Fast R-CNN: Region of Interest Pooling



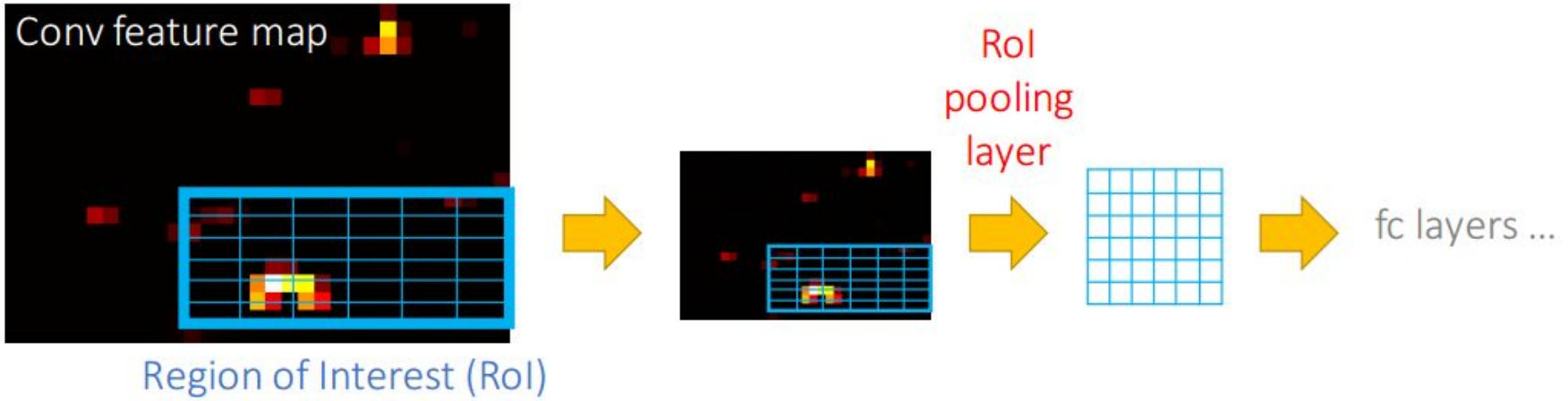
Fast R-CNN: Region of Interest Pooling



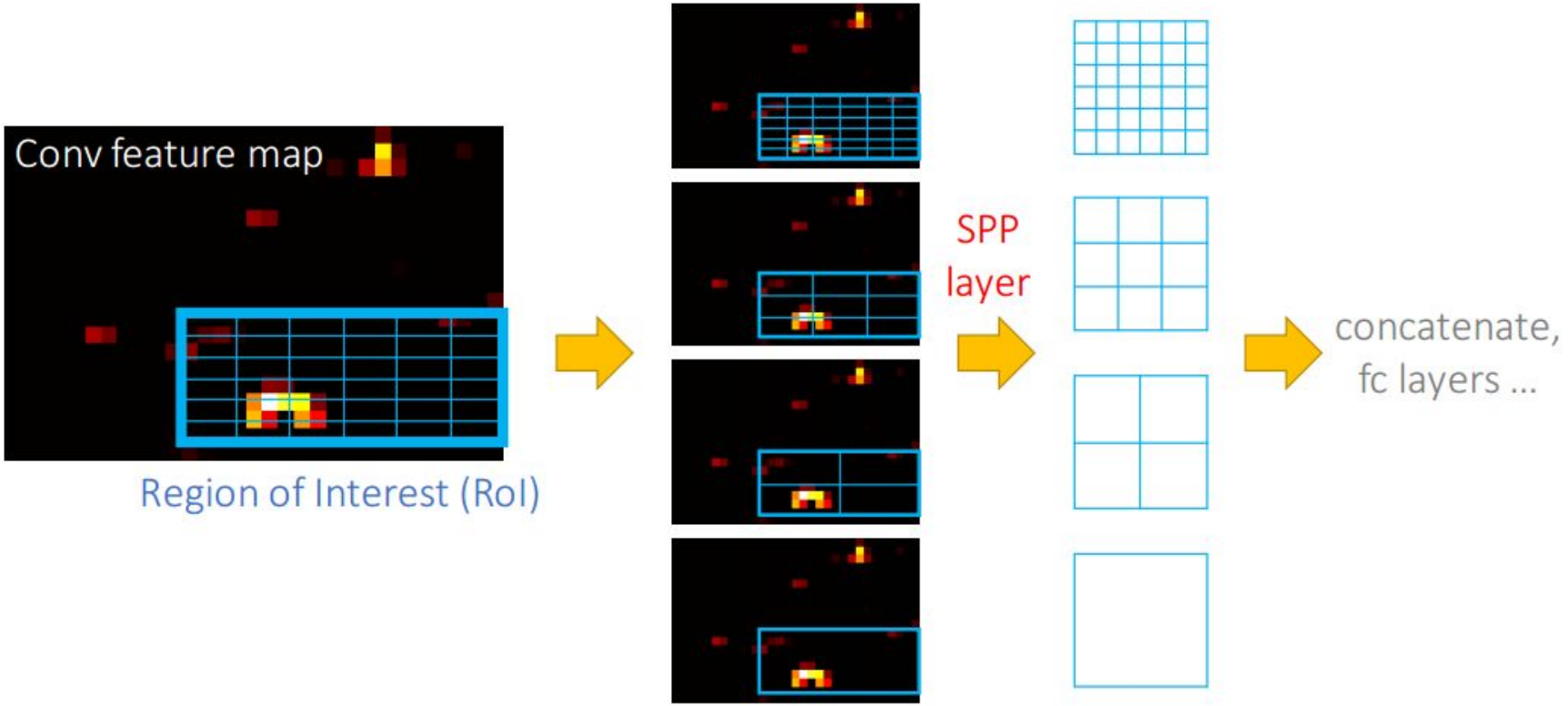
Fast R-CNN: Region of Interest Pooling



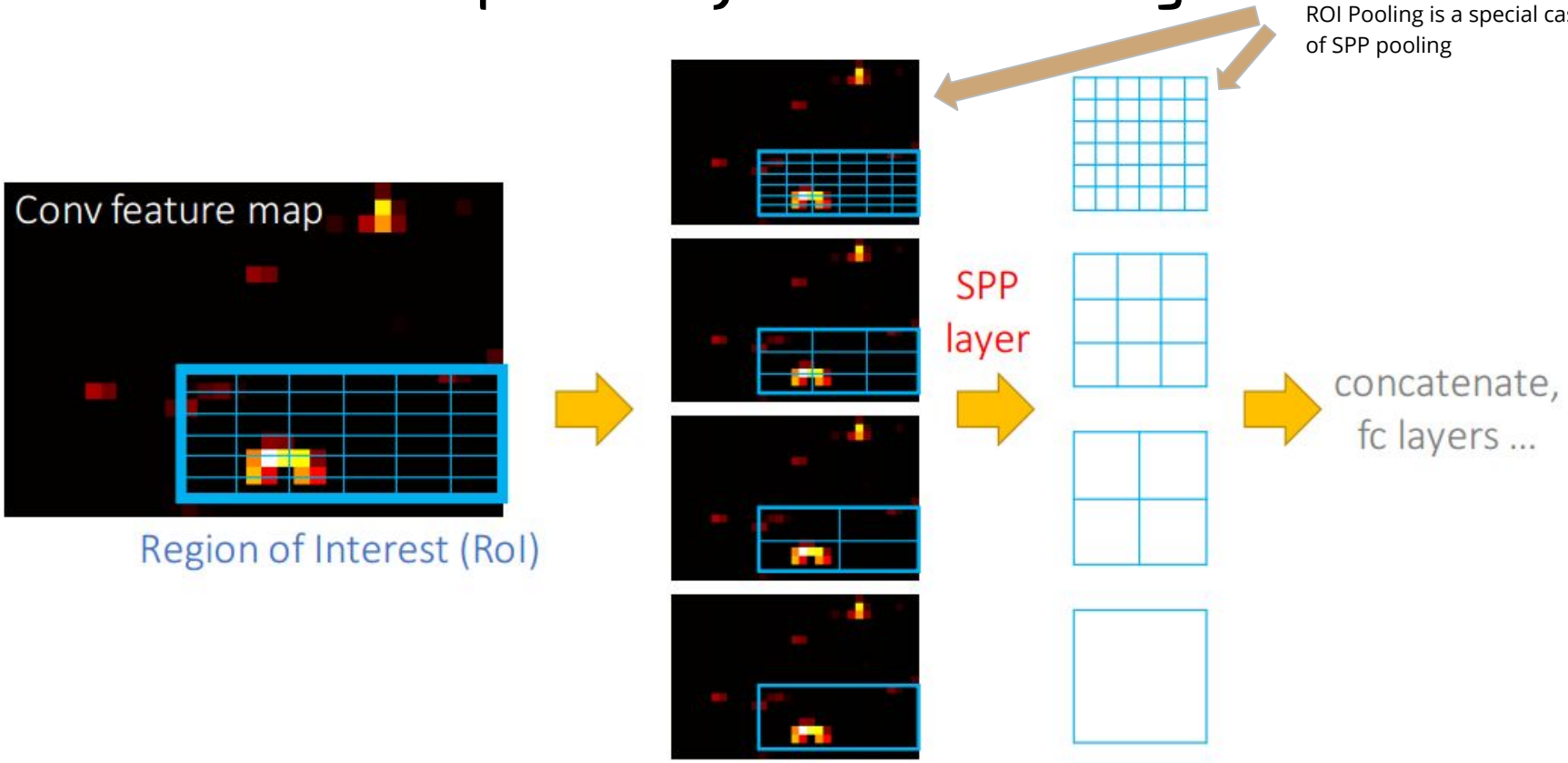
Fast R-CNN: Region of Interest Pooling



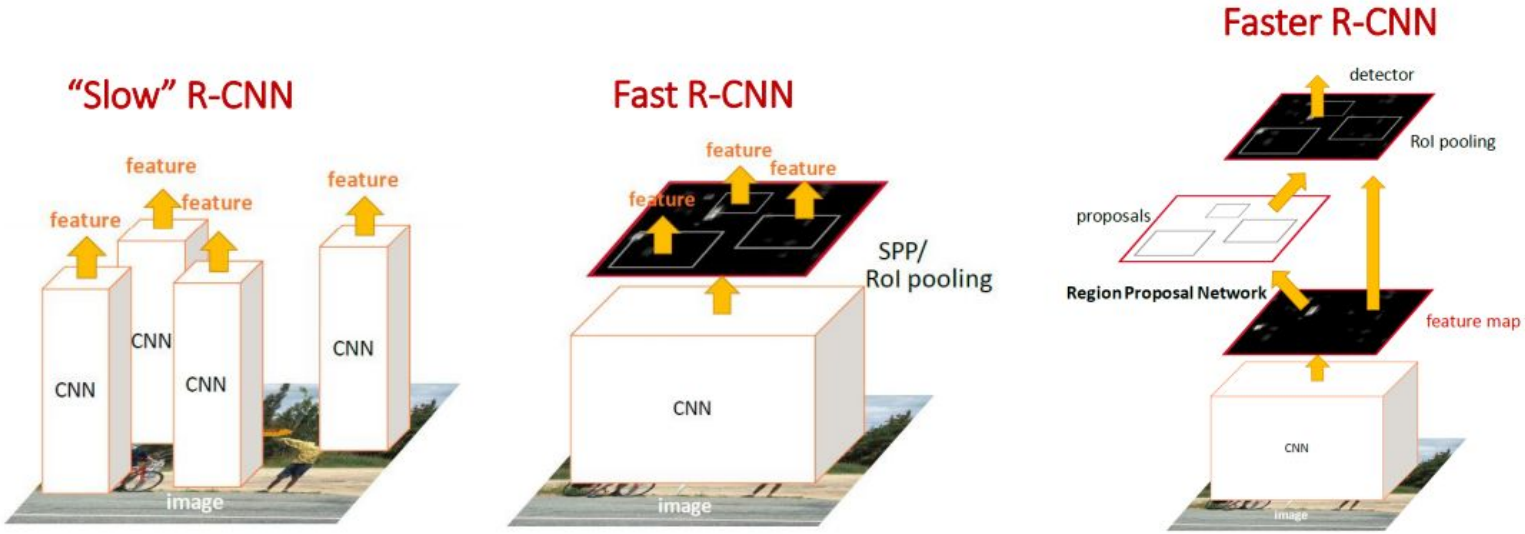
Fast R-CNN: Spatial Pyramid Pooling



Fast R-CNN: Spatial Pyramid Pooling



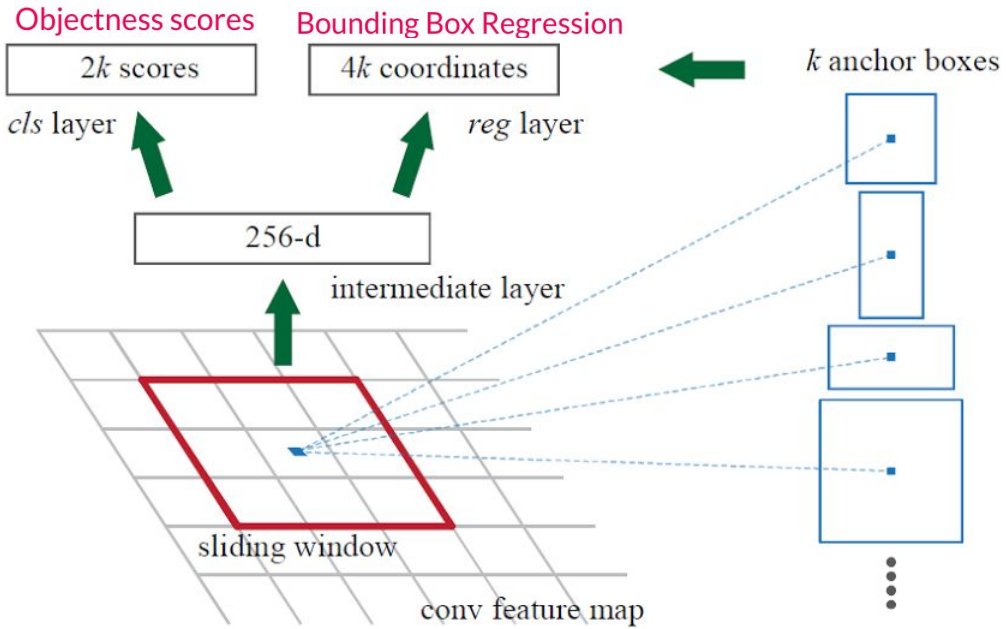
Faster R-CNN



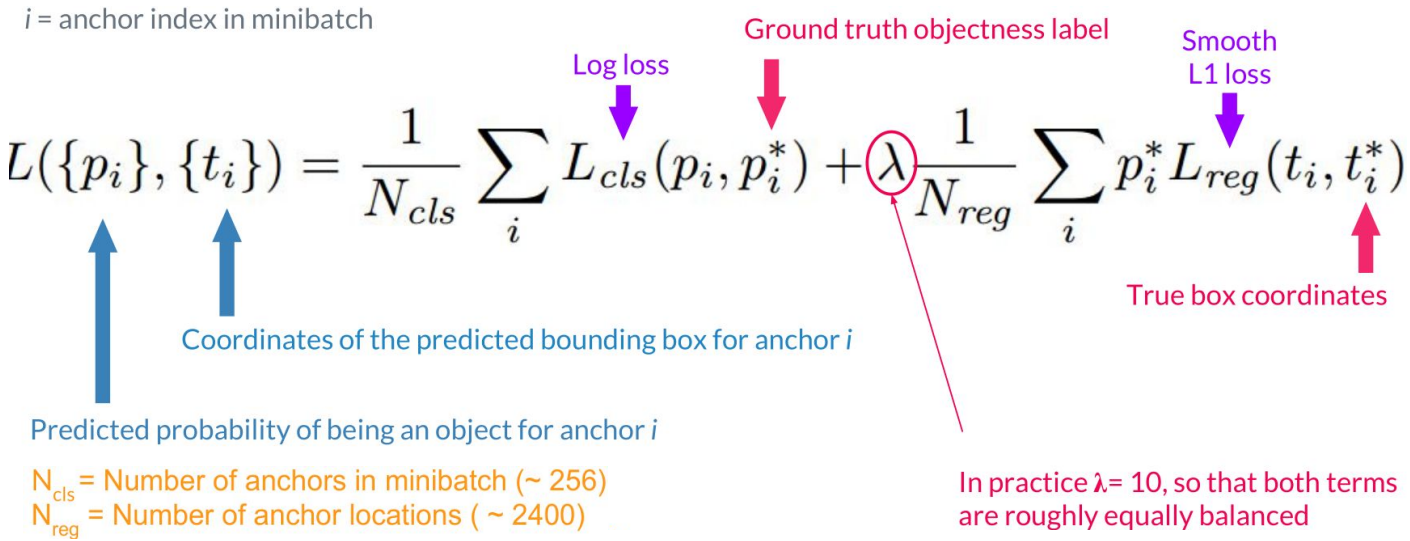
- In R-CNN and Fast R-CNN, bottleneck is 'region proposal' (selective search, CPMC, MCG).
- Replace external proposals with Region Proposal Network (RPN), trained with model directly.
- After RPN use ROI Pooling like Fast R-CNN

Region Proposal Network

- Slide a small window (anchor boxes) on the feature map.
- Have multiple such (k) anchor boxes.
- Build small network for
  - binary classification of object in an anchor. (👍)
  - regression of bounding box (anchor->proposal)
- Position of sliding window provides localization information with reference to the image.
- Binary classification gives probability that each anchor shows an object.
- Regression score gives offset from anchor boxes
- Anchors are selected for different scales and different aspect ratio, generally 3 each, total 9.



Faster R-CNN: RPN



Anchor is labelled positive if:

- anchor is the one with highest IoU overlap with ground truth box
- anchor has IoU overlap with a ground-truth box higher than 0.7

Anchor is labelled negative if IoU lower than 0.3 for all ground-truth boxes.

50%/50% ratio of positive to negative samples

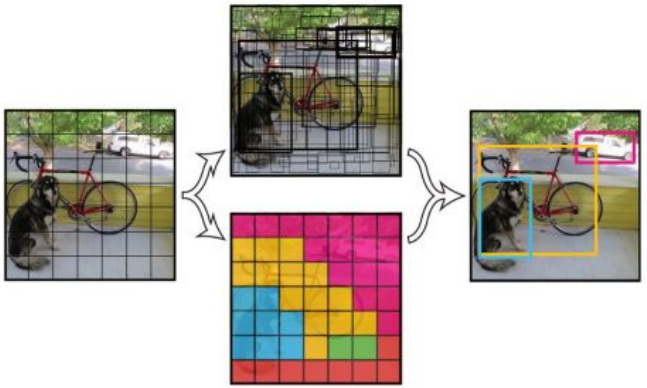
Source: A Salvador

Faster R-CNN: Comparison accuracy/speed

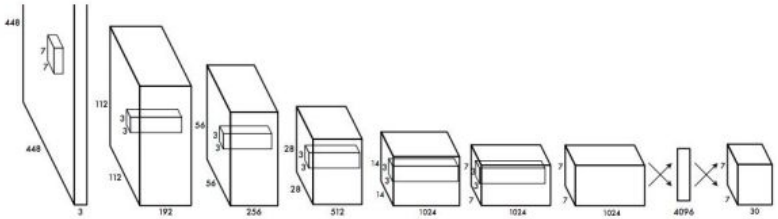
	R-CNN	Fast R-CNN	**Faster R-CNN**
Test time per image	50 seconds	2 seconds	0.2 seconds
Speedup (+proposal)	1x	25x	250x
mAP (VOC 2007)	66.0	66.9	66.9

You Only Look Once (YOLO)

- Divide image into S x S grid
- Within each grid cell predict:
  - B Boxes (4 coordinates + confidence)
  - Class scores: C scores
- Regression from image to 7 x 7 x (5 * B + C) tensor
- Direct prediction using a CNN
- Faster than Faster R-CNN but not as good
- Struggles with small objects and different aspect ratio



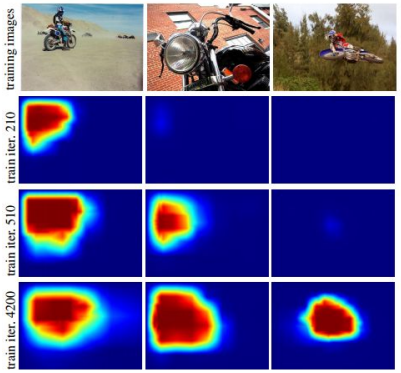
Real-Time Detectors	Train	mAP	FPS
100Hz DPM [30]	2007	16.0	100
30Hz DPM [30]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [37]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[27]	2007+2012	73.2	7
Faster R-CNN ZF [27]	2007+2012	62.1	18



Can you do this without spatial ground-truth?

Weakly supervised object localization

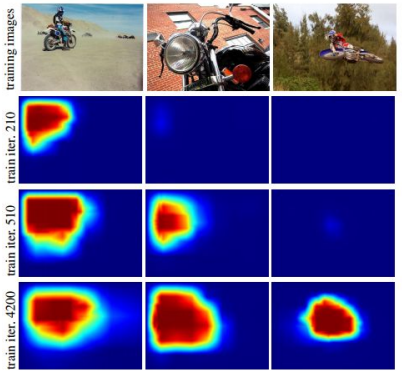
- If you look at the pixels which give maximum activation for a given class, it is no surprise that pixels corresponding to actual object have higher activation.
- But how do you control of actual object pixels? Remember the ‘tank problem’?
- Also this is not reliable in a cluttered scene containing many objects, and partially occluded objects or cropped objects.
- Oquab et al,
    - removed the Fully Connected layer at the end of a CNN model to prevent losing any more localization information and,
    - Added global max-pooling (GMP) to output layer to find highest scoring object position in the image.

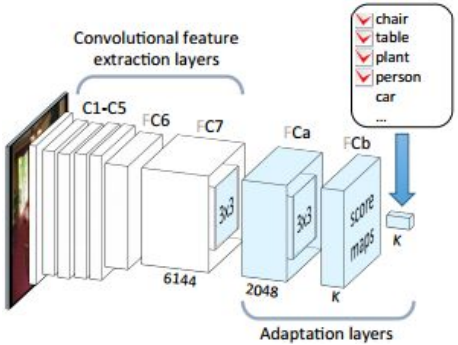


Can you do this without spatial ground-truth?

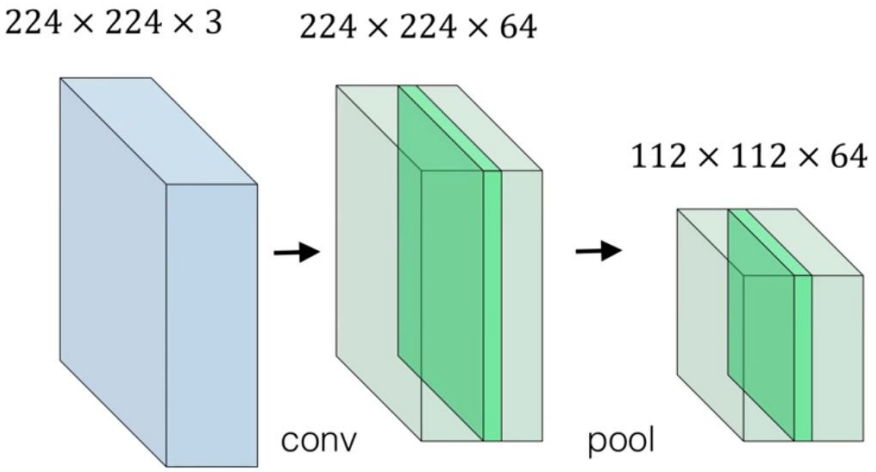
Weakly supervised object localization

- If you look at the pixels which give maximum activation for a given class, it is no surprise that pixels corresponding to actual object have higher activation.
- But how do you control of actual object pixels? Remember the ‘tank problem’?
- Also this is not reliable in a cluttered scene containing many objects, and partially occluded objects or cropped objects.
- Oquab et al,
  - removed the Fully Connected layer at the end of a CNN model to prevent losing any more localization information and,
  - Added global max-pooling (GMP) to output layer to find highest scoring object position in the image.
- GMP enforces correspondence between feature maps and categories, and thus they can be interpreted as ‘confidence maps’.
- Best part - it is parameter free.

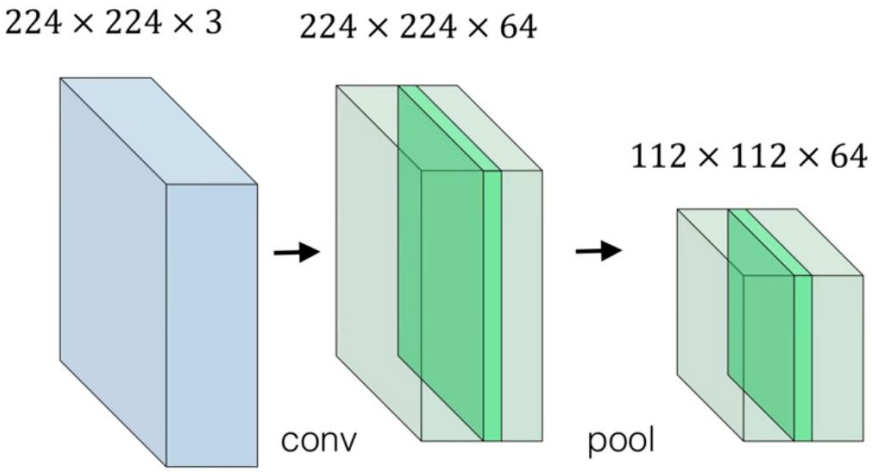


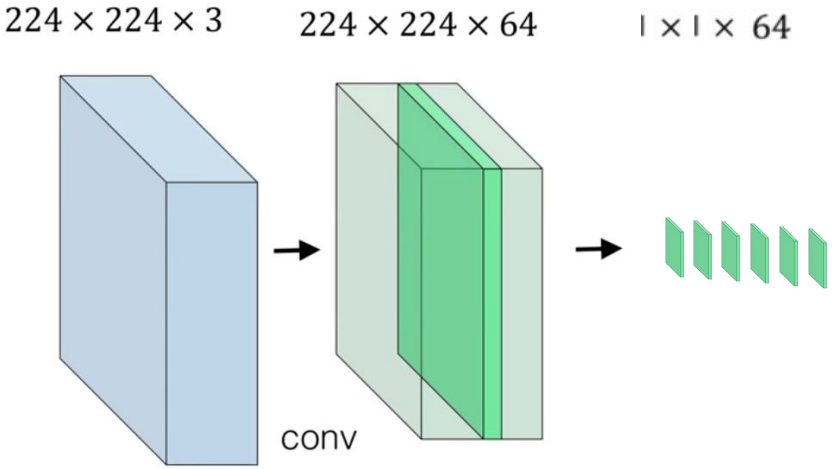


Max Pooling vs Global Max Pooling

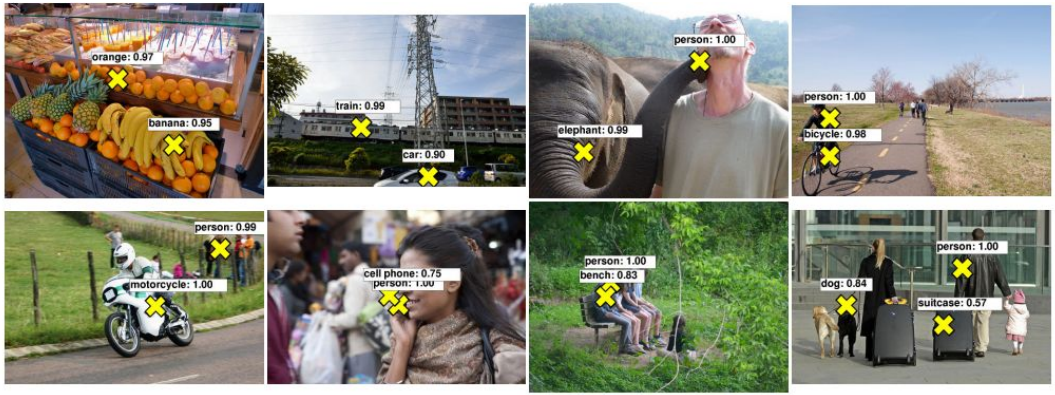


Max Pooling vs Global Max Pooling





Global Max Pooling - Results



Object-level sup.	mAP
A. NUS-SCM [51]	82.2
B. OQUAB [37]	82.8
Image-level sup.	**mAP**
C. Z&F [60]	79.0
D. CHATFIELD [6]	83.2
E. NUS-HCP [56]	84.2
F. FULL IMAGES	78.7
G. WEAK SUP	**86.3**

- Localization is only about finding ‘center’ point, cannot draw bounding box
- Helps improve the classification score
- Masked pooling is where you minimize the object score outside the masked area (requires location GT)

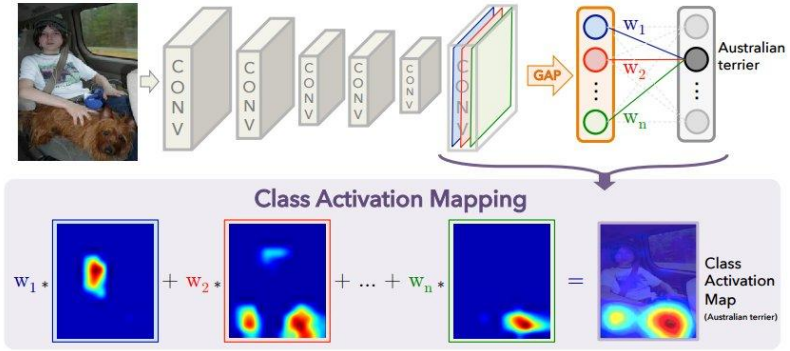
Setup	Classification		Location Prediction	
Dataset	VOC	COCO	VOC	COCO
H. FULL IMAGES	76.0	51.0	-	-
I. MASKED POOL	**82.3**	62.1	72.3	**42.9**
J. WEAK SUP	81.8	**62.8**	74.5	41.2
K. CENTER PRED.	-	-	50.9	19.1
L. RCNN*	79.2	-	**74.8**	-

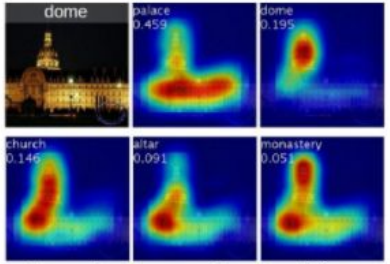
Class activation map

- Replace Global Max Pooling (GMP) with Global Average Pooling (GAP).
- The use of average pooling encourages the network to identify the complete extent of the object
- Intuition behind this is that the loss for average pooling benefits when the network identifies all discriminative regions of an object as compared to max pooling
- Class activation map is when you take the output of GAP and add a ‘fully connected layer’ (without nonlinearity). Use the weights of this layer to weight the activation at unit k, to generate a spatial map.

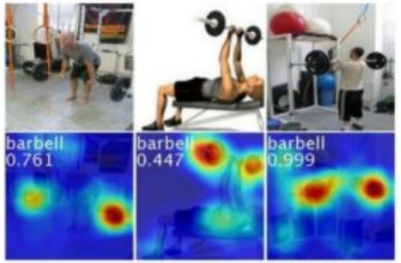
$$M_c(x,y) = \sum_{d \,\epsilon\, \mathbf{D}} w_d^c \, f_d(x,y) \quad \text{where } w_d^c \text{ is the learned of class } c \text{ for feature map } d$$

- For training, you maximize the cross entropy between predicted class and true distribution $P(c) = \frac{\exp(\sum_{xy} M_c(x,y))}{\sum_c \exp(\sum_{xy} M_c(x,y))}$
- This means it can “highlight” only one salient class in a given image. Gives more than center but not bounding box.



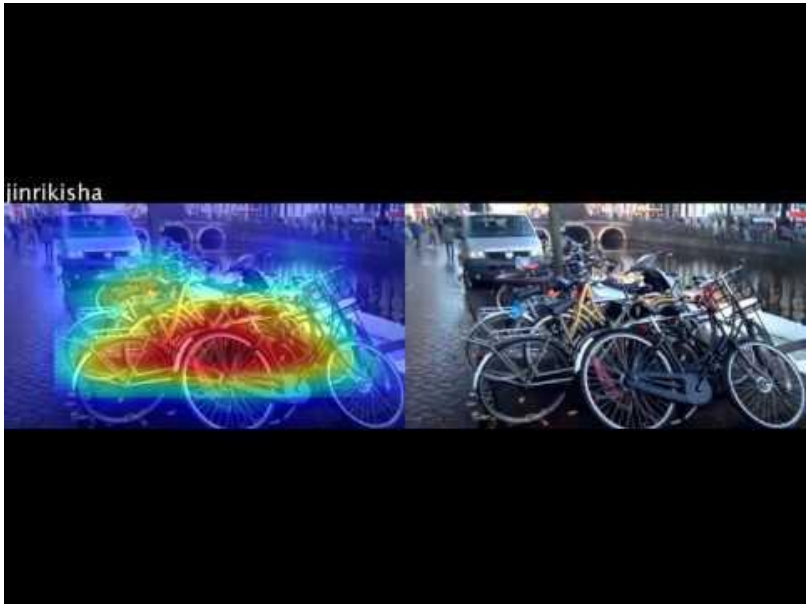


Class activation maps of top 5 predictions



Class activation maps for one object class

Class activation map



References

1. Tools for efficient Object Detection [pdf]
2. R-CNN for Object Detection [pdf]
3. Segmentation as Selective Search (Poster) [pdf]
4. Object Detection, Lecture at UPC [pdf]
5. Stanford CS231n Lecture 8 slides [pdf]
6. Faster R-CNN: Towards real-time object detection [pdf]
7. Is localization for free? [pdf]
8. Learning Deep Features for Discriminative Localization [pdf]

Questions?