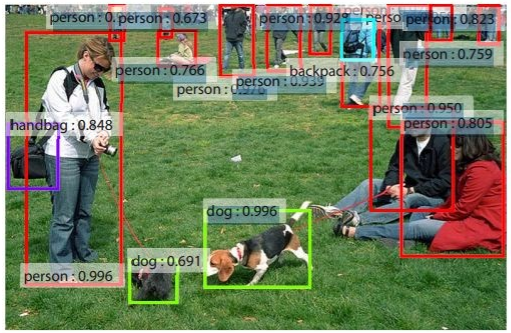
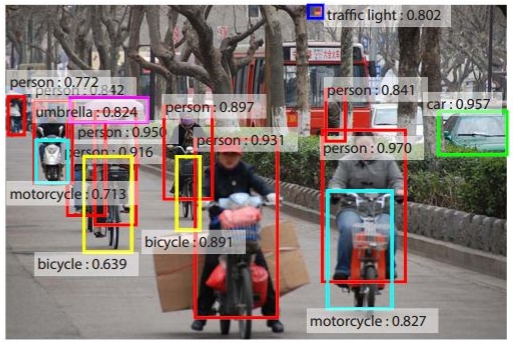
CSCM77
Object Detection

Prof. Xianghua Xie
x.xie@swansea.ac.uk
http://csvision.swan.ac.uk

Localisation and detection



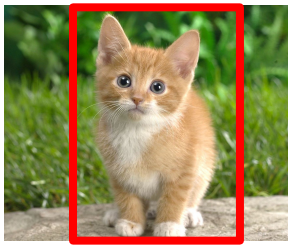


Results from Faster R-CNN, Ren et al 2015

Classification + Localisation

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

CAT

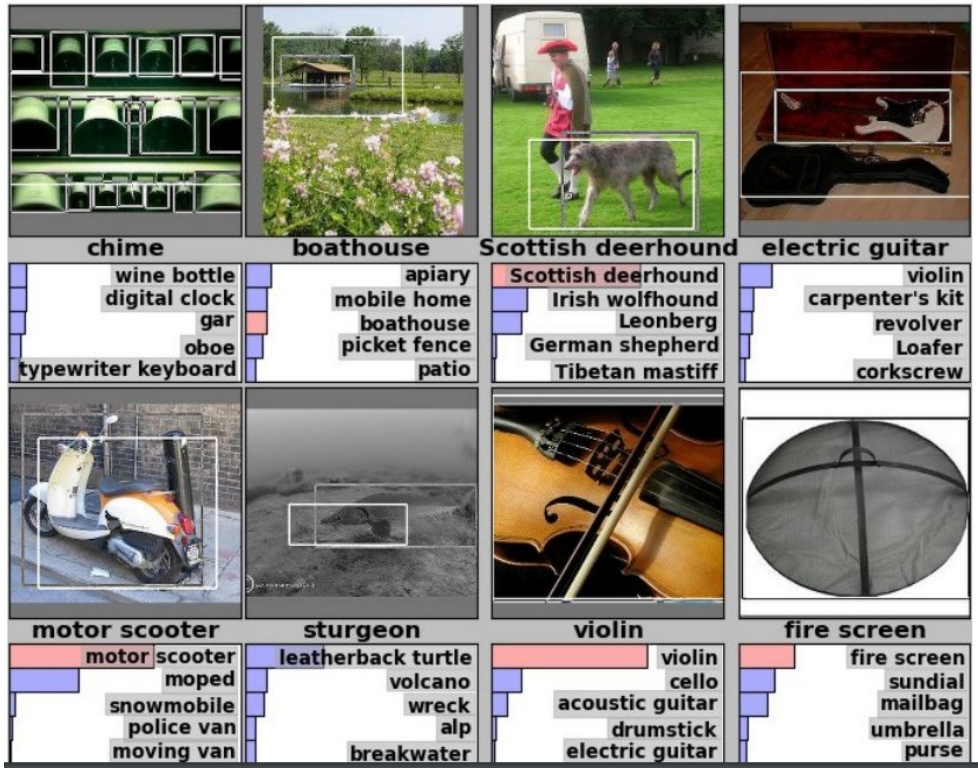


(x, y, w, h)

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

Classification + Localisation

- ImageNet
  - 1000 classes
  - Each image has 1 class, at least one bounding box
  - ~800 training images per class
  - Algorithm produces 5 (class, box) guesses
  - Example is correct if at least one one guess has correct class AND bounding box at least 0.5 intersection over union (IoU)



Krizhevsky et. al. 2012

Localisation using Regression

Input: image

Only one object, simpler than detection

Neural Net

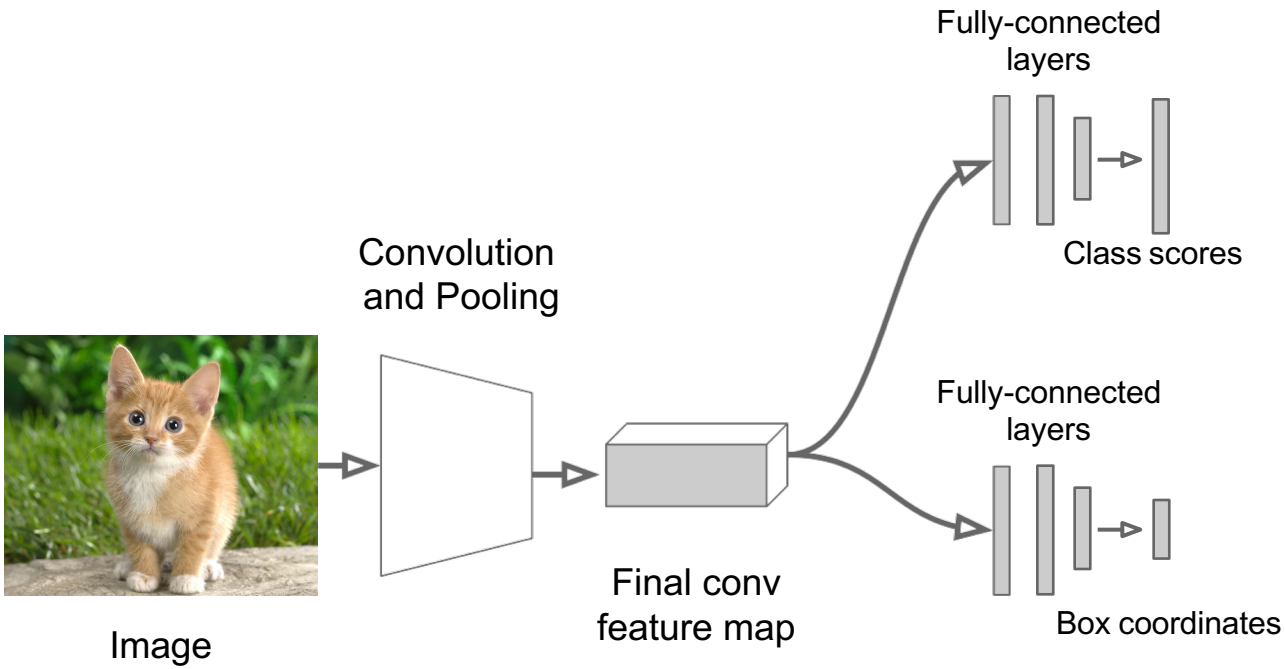
Output: Box coordinates (4 numbers)

Correct output: box coordinates (4 numbers)

Loss: L2 norm

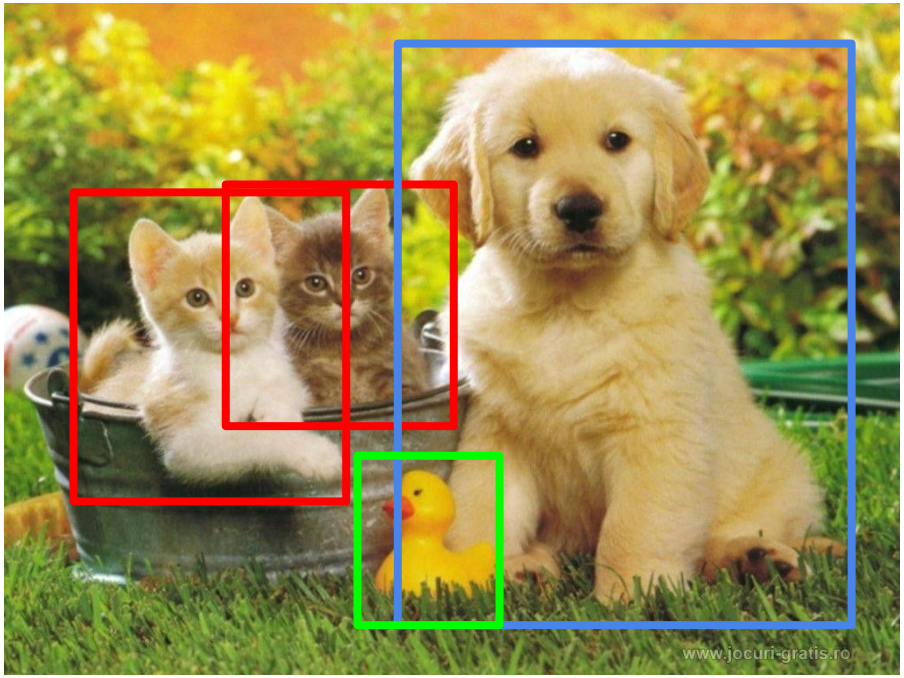
Classification + Localisation

- Use shared CNN feature for classification and localisation



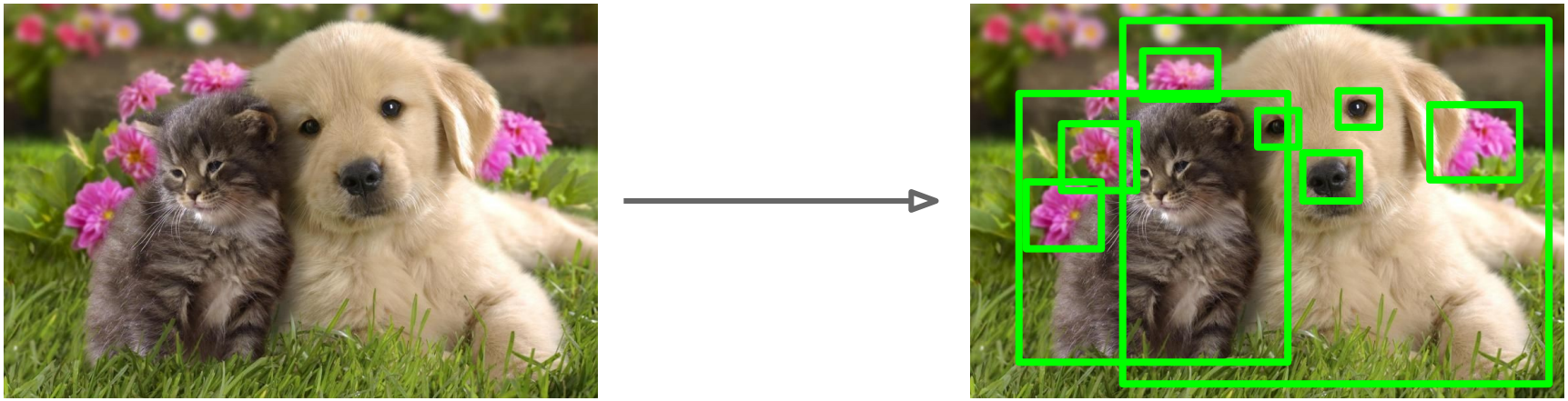
Object Detection

- Need to test many possible positions and scales: numerous combinations
- Class specific
- Evaluate only a small subset



Region proposals

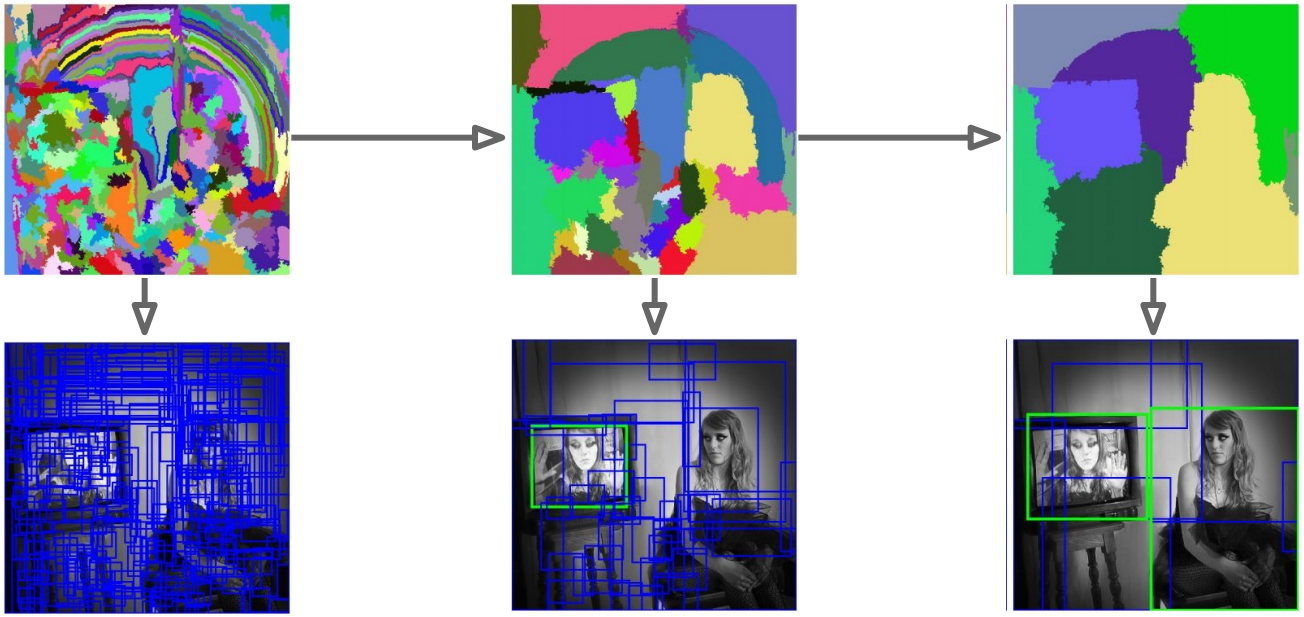
- Find “blobby” image regions that are likely to contain objects
- “Class-agnostic” object detector, i.e. foreground detector
- Look for “blob-like” regions



Region proposals

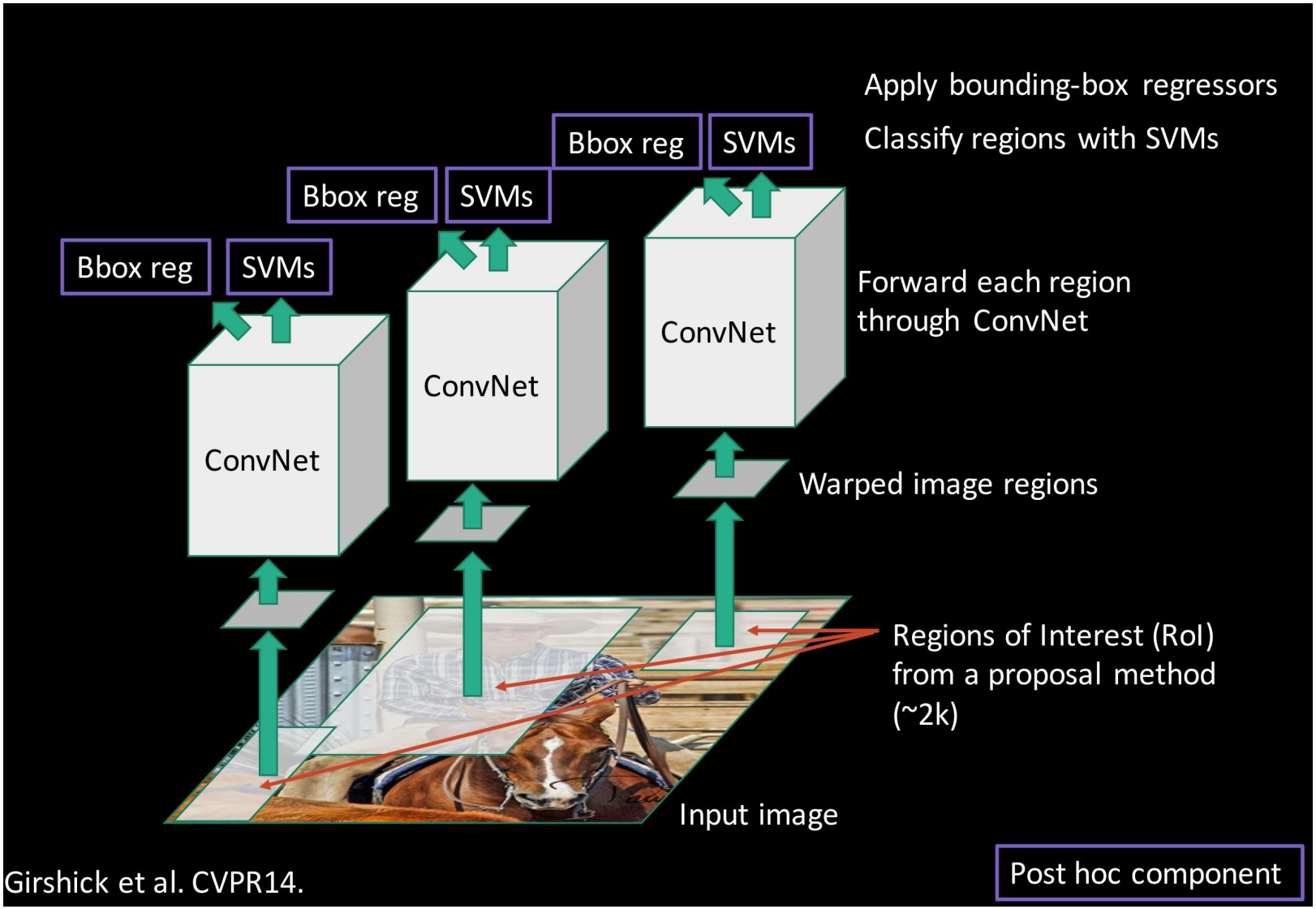
- Selective search, e.g.
  - Bottom-up segmentation, merging regions at multiple scales

Convert regions to boxes



Uijlings et al, “Selective Search for Object Recognition”, IJCV 2013

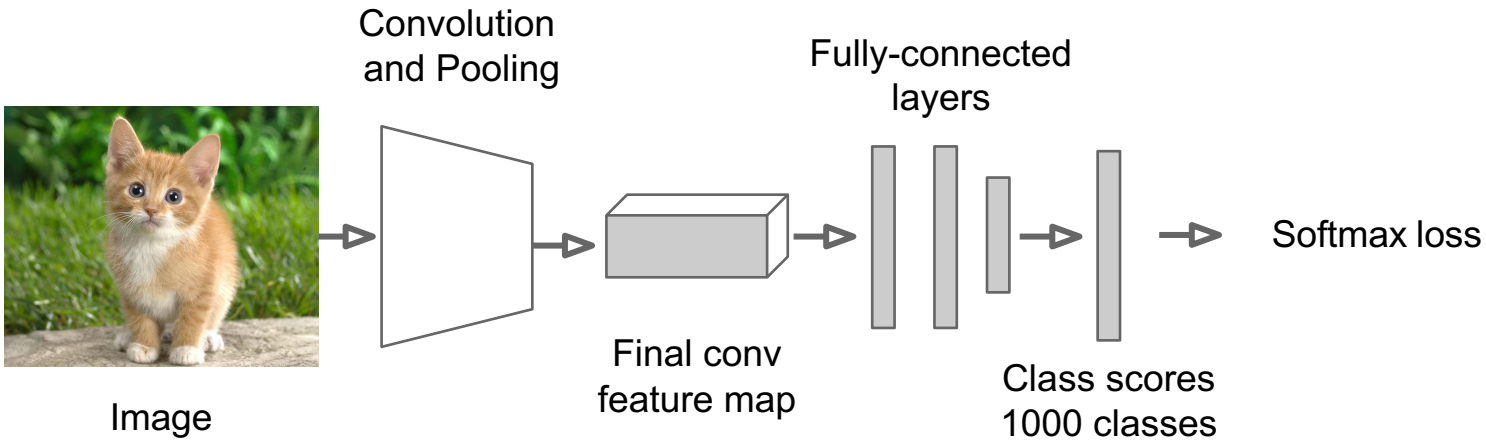
R-CNN: region proposal CNN



Girschick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014

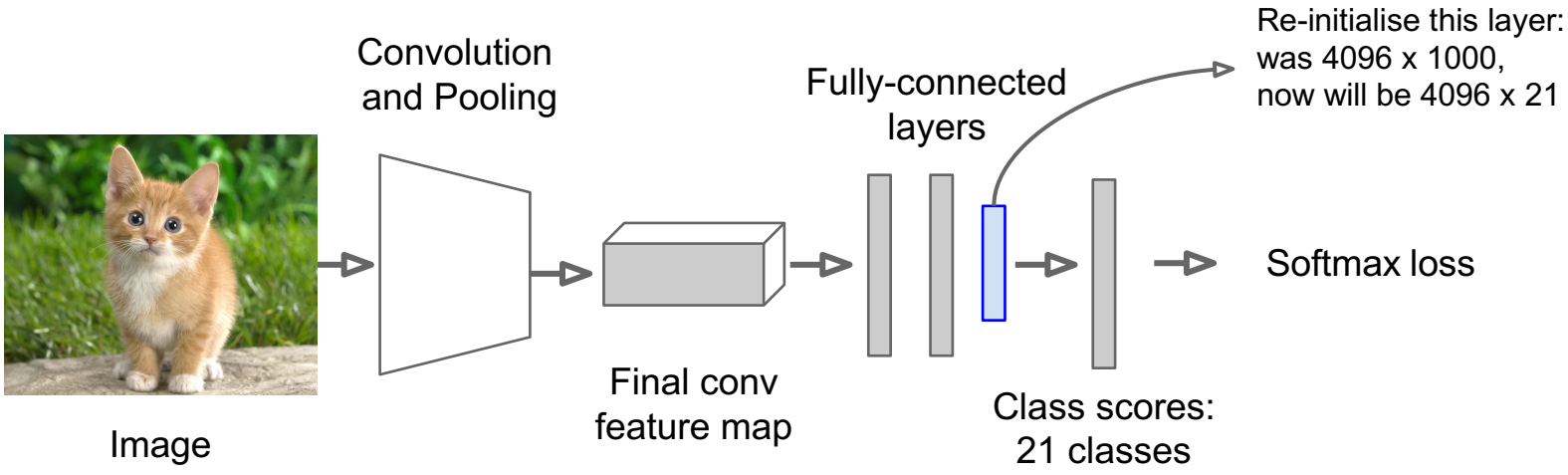
R-CNN training

- Step 1: Train a classification model for ImageNet (AlexNet)



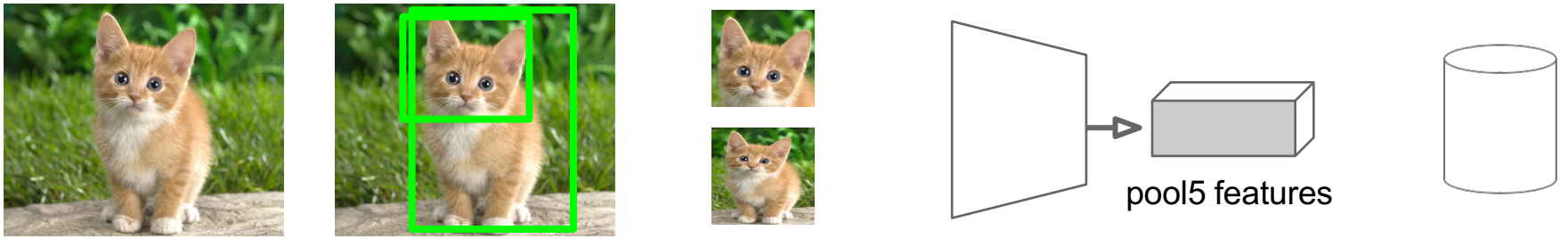
R-CNN training

- Step 2: Fine-tune model for detection
  - Instead of 1000 ImageNet classes, want 20 object classes + background
  - Throw away final fully-connected layer, reinitialise from scratch
  - Keep training model using positive / negative regions from detection images



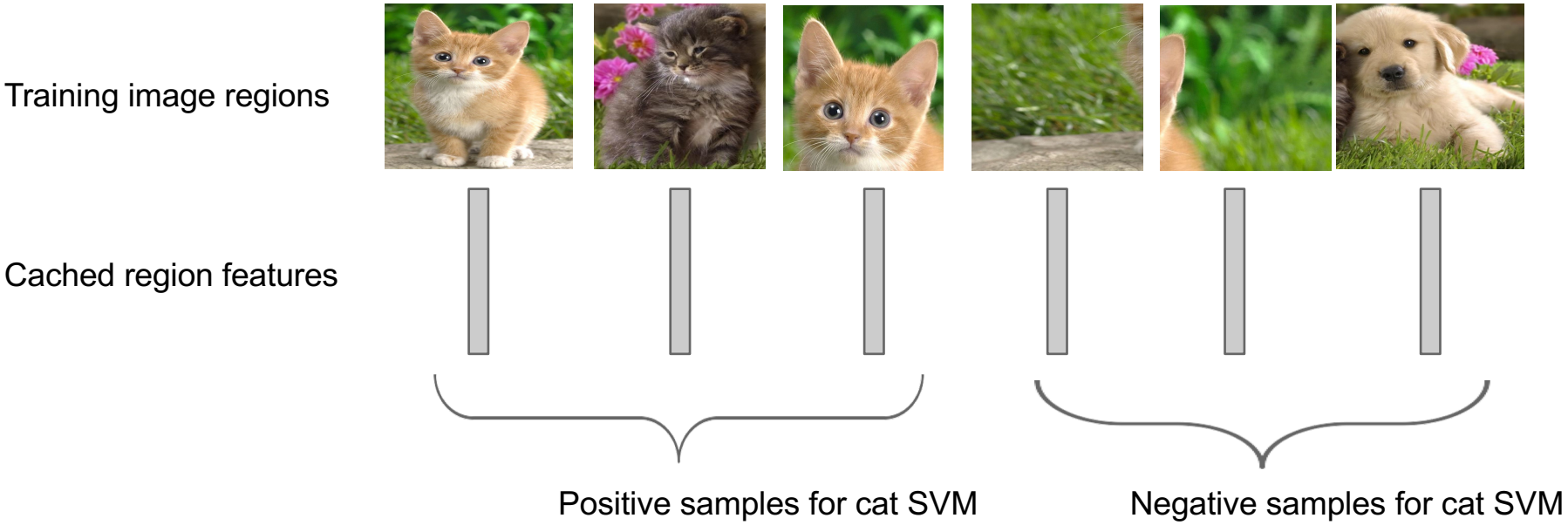
R-CNN training

- Step 3: Extract features
  - Extract region proposals for all images
  - For each region: warp to CNN input size, run forward through CNN, save pool5 features to disk (large amount of data)



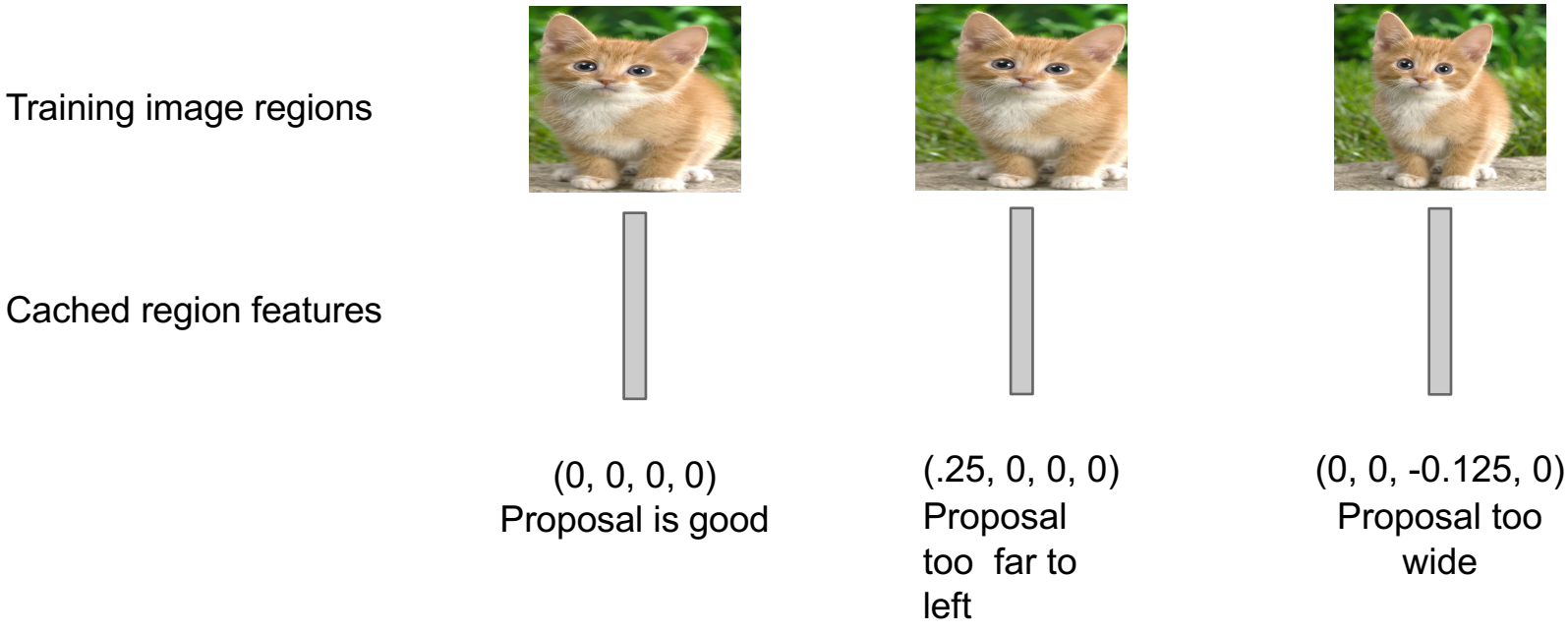
R-CNN training

- Step 4: Train one binary SVM per class to classify region features



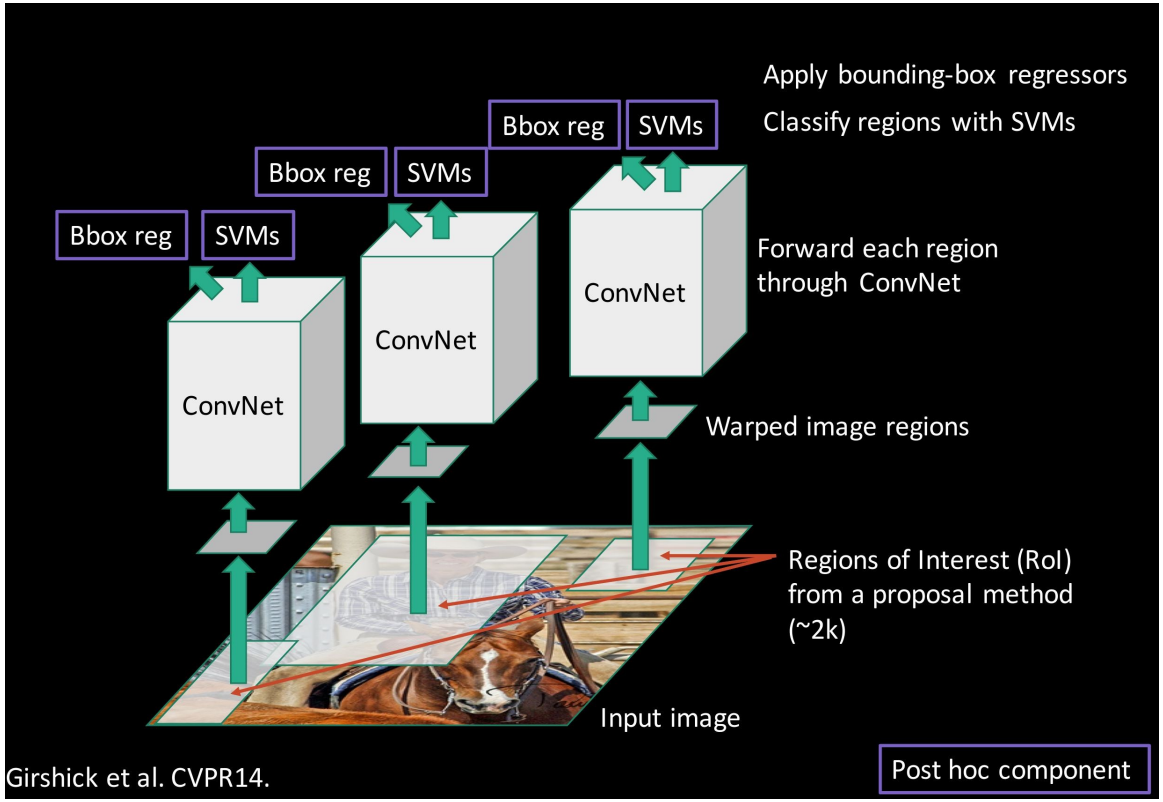
R-CNN training

- Step 5 (bbox regression): For each class, train a linear regression model to map from cached features to offsets to ground-truth (GT) boxes to make up for "slightly wrong" proposals



R-CNN: region proposal CNN

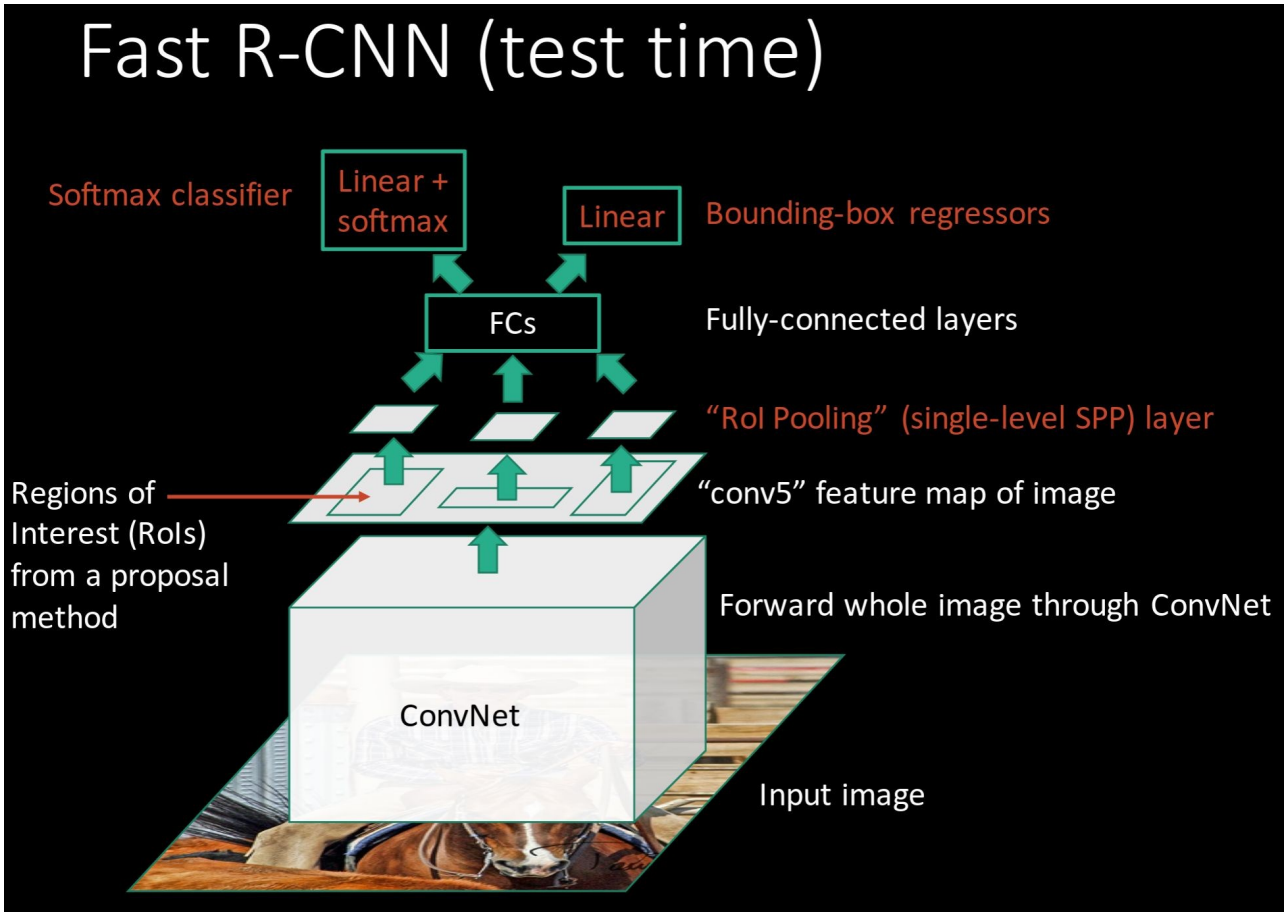
- Slow at test-time: need to run full forward pass of CNN for each region proposal
- SVMs and regressors are post-hoc: CNN features not updated in response to SVMs and regressors
- Complex multistage training pipeline



Girshick et al. CVPR14.

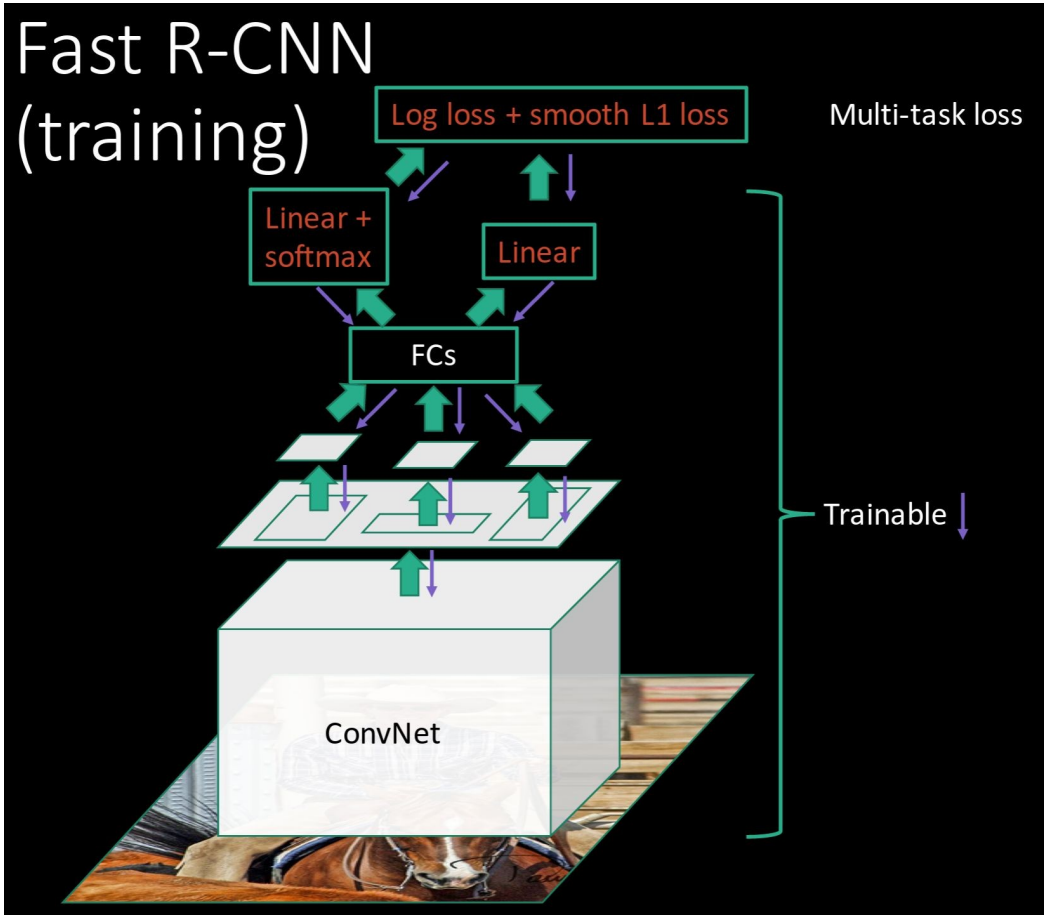
Fast R-CNN

- R-CNN Problem #1: Slow at test-time due to independent forward passes of the CNN
  - Solution: Share computation of convolutional layers between proposals for an image



Fast R-CNN

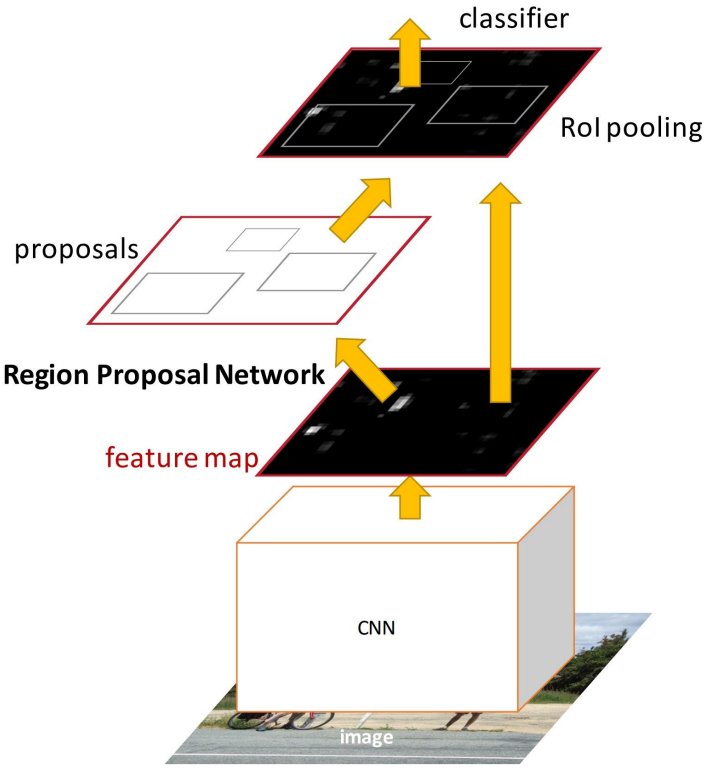
- R-CNN Problem #2: Post-hoc training: CNN not updated in response to final classifiers and regressors
  - Solution: Just train the whole system end-to-end all at once!
  - This also reduce training complexity (R-CNN problem #3)



Faster R-CNN

- Fast R-CNN requires external region proposal

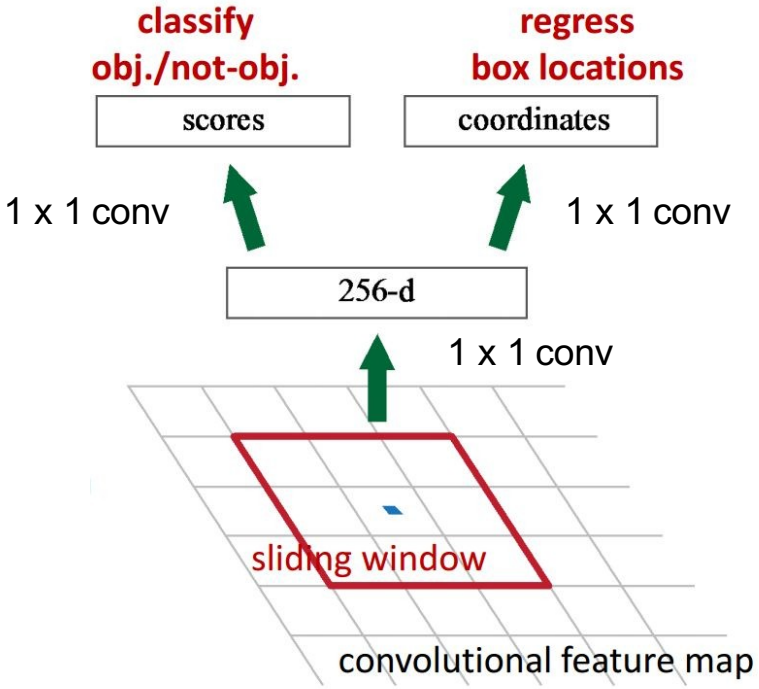
- Faster R-CNN
  - Insert a Region Proposal Network (RPN) after the last convolutional layer
  - RPN trained to produce region proposals directly; no need for external region proposals
  - After RPN, use RoI Pooling and an upstream classifier and bbox regressor just like Fast R-CNN



Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

Faster R-CNN

- Region proposal network
  - Sliding window on the feature map
  - Build a small network for:
    - classifying object or not-object, and
    - regressing bbox locations
  - Position of the sliding window provides localization information with reference to the image
  - Box regression provides finer localisation information

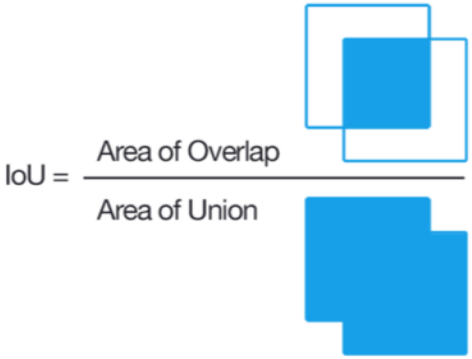


Results

	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**

mAP: mean average precision

- A detection is a true positive if it has IoU (intersection over union) with a ground-truth box greater than some threshold (usually 0.5)
- Combine all detections from all test images to draw a precision/recall curve for each class; AP is area under the curve
- Compute average precision (AP) separately for each class, then average over classes



$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$