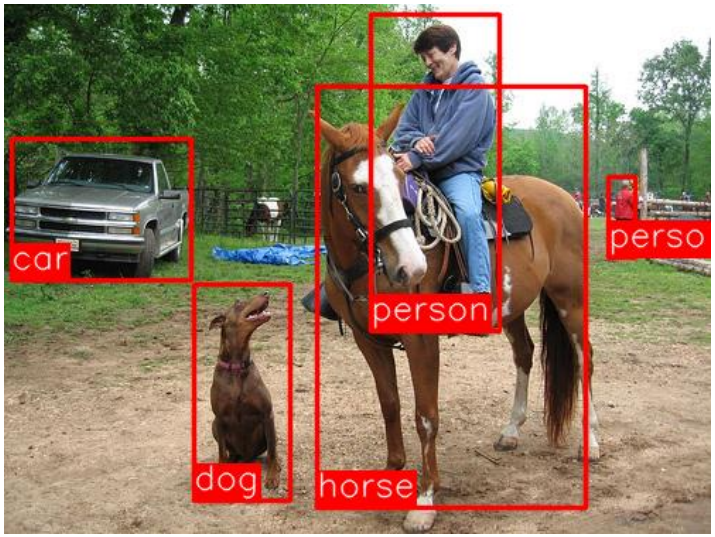
Object detection and segmentation

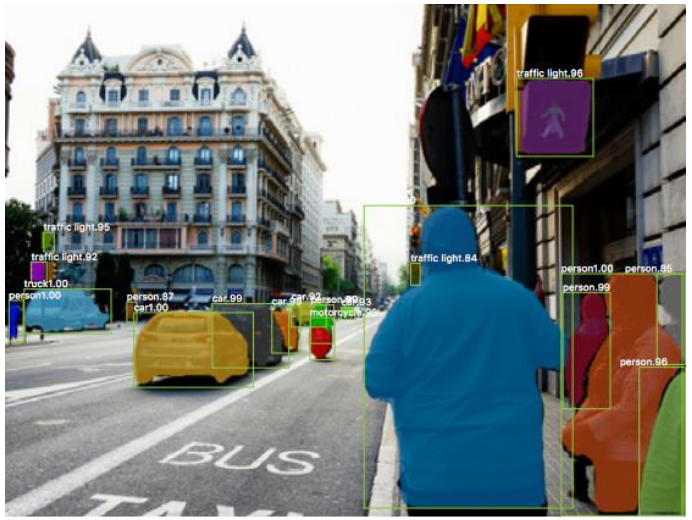
ISTD 50.035

Computer Vision

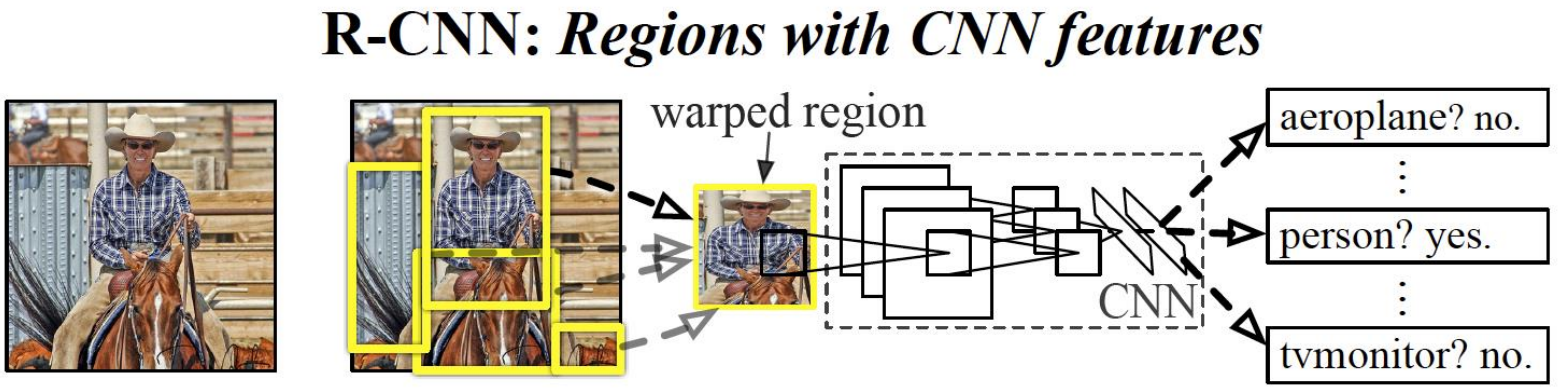
Object detection / segmentation

- Finding different objects in an image and classify them

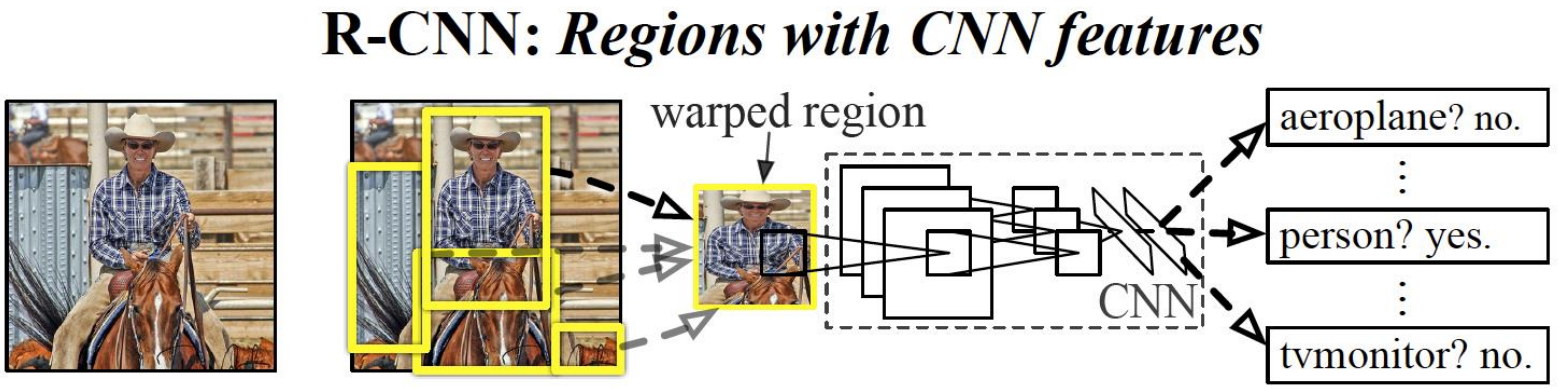




Issue with R-CNN



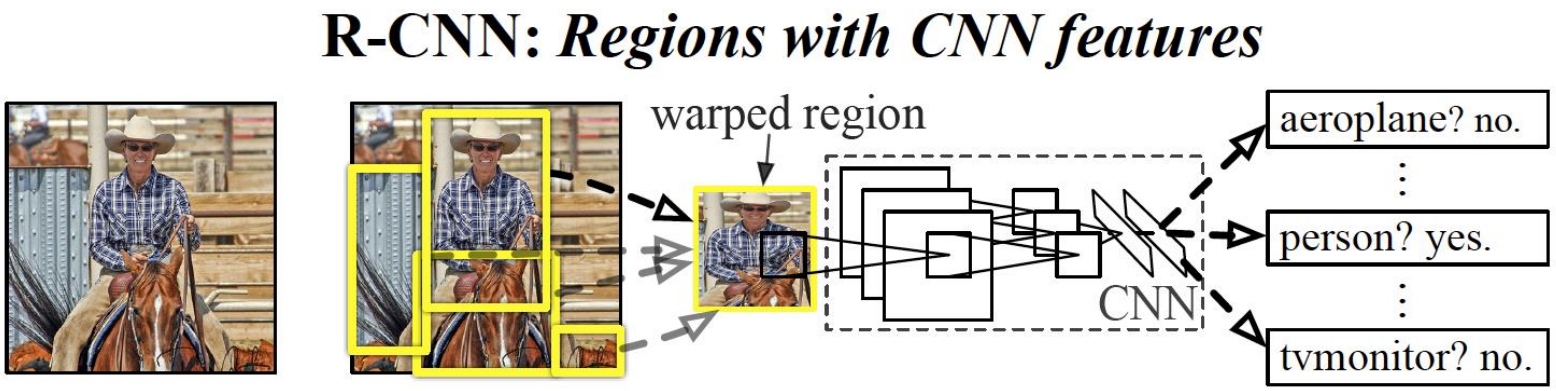
Issue with R-CNN



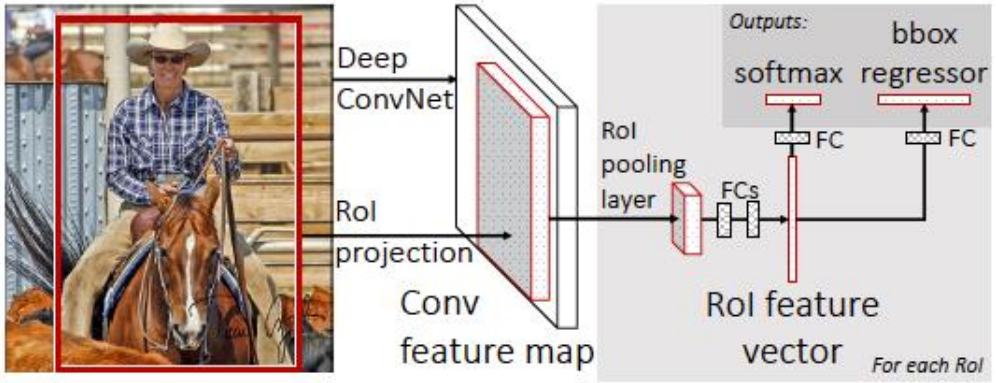
- Independent forward pass of the CNN for every region proposal: independent feature computation (~2000 proposals per image)

- Train three different systems separately: feature extraction (domain specific fine-tuning), classifier (SVM) and bounding box regression

Fast R-CNN



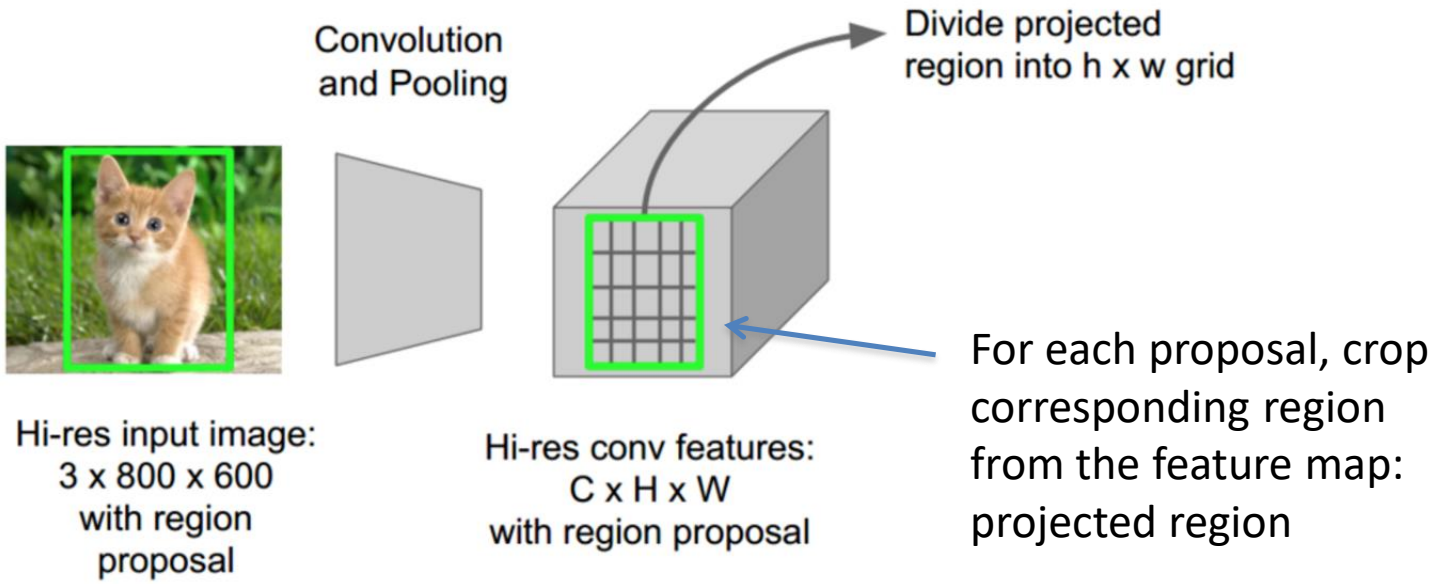
Input: image and region proposal (bounding boxes from selective search)



[Ross Girshick; 2015]

- Key ideas:
  - Single pass of CNN to extract feature
  - For each proposal, crop corresponding region from the feature map

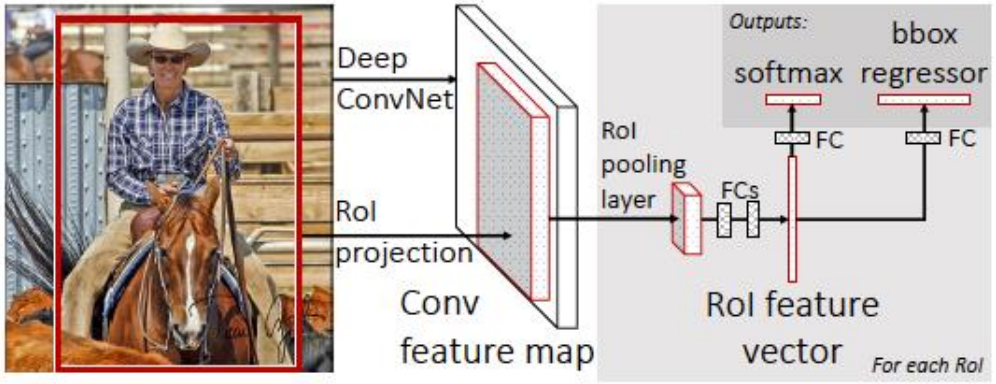
ROI (Region of Interest) Pooling



- Need to convert arbitrary projected region into fixed spatial size (hxw) for softmax and regression: ROI pooling
- Divide into hxw grid (e.g. 5x5 in this example)
- max pooling in each grid of size H/h x W/w
- Pooling is applied independently for each feature map channel
- Compare to ‘standard’ pooling: variable size window

Fast R-CNN

Input: image and region proposal (bounding boxes from selective search)

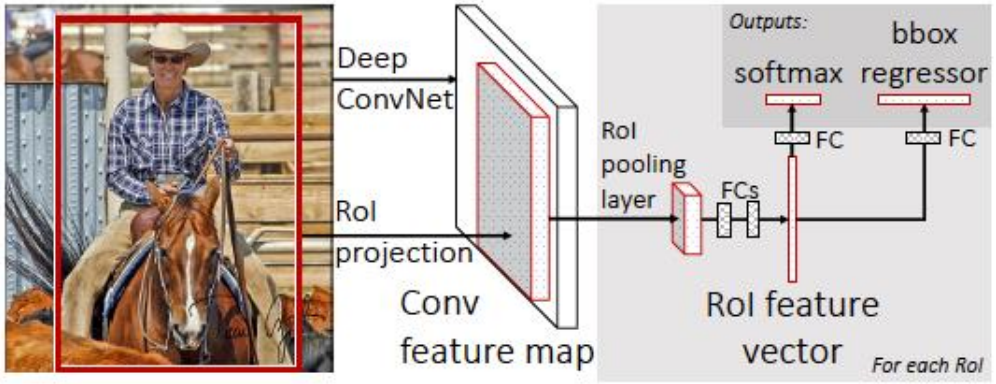


[Ross Girshick; 2015]

- Train feature extraction, classifier and regression at the same time
- For each ROI, two outputs:
  - Probability over K+1 classes (K object types and background)
  - Bounding box parameters (4 parameters same as R-CNN)
- Supervised training: For each ROI, labeled with
  - Ground-truth class label u
  - Ground-truth bounding-box regression target v
  - Loss function: classification loss (w.r.t. u) and bounding-box loss (w.r.t. v)

Multi-task loss

Input: image and region proposal (bounding boxes from selective search)



[Ross Girshick; 2015]

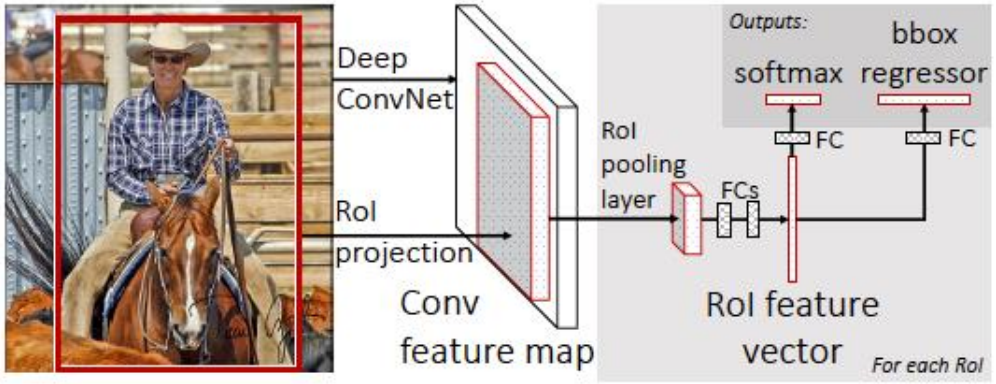
- Supervised training: For each ROI, labeled with
  - Ground-truth class label u
  - Ground-truth bounding-box regression target v
  - Loss function: classification loss (w.r.t. u) and bounding-box loss (w.r.t. v)

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

Classification loss (K+1 classes): cross entropy $L_{\text{cls}}(p, u) = -\log p_u$

Multi-task loss

Input: image and region proposal (bounding boxes from selective search)



[Ross Girshick; 2015]

For background, u=0, no bbox regression loss
[u>=1] = 1 if u>=1, 0 otherwise

$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{\text{x,y,w,h}\}} \text{smooth}_{L_1}(t_i^u - v_i),$$

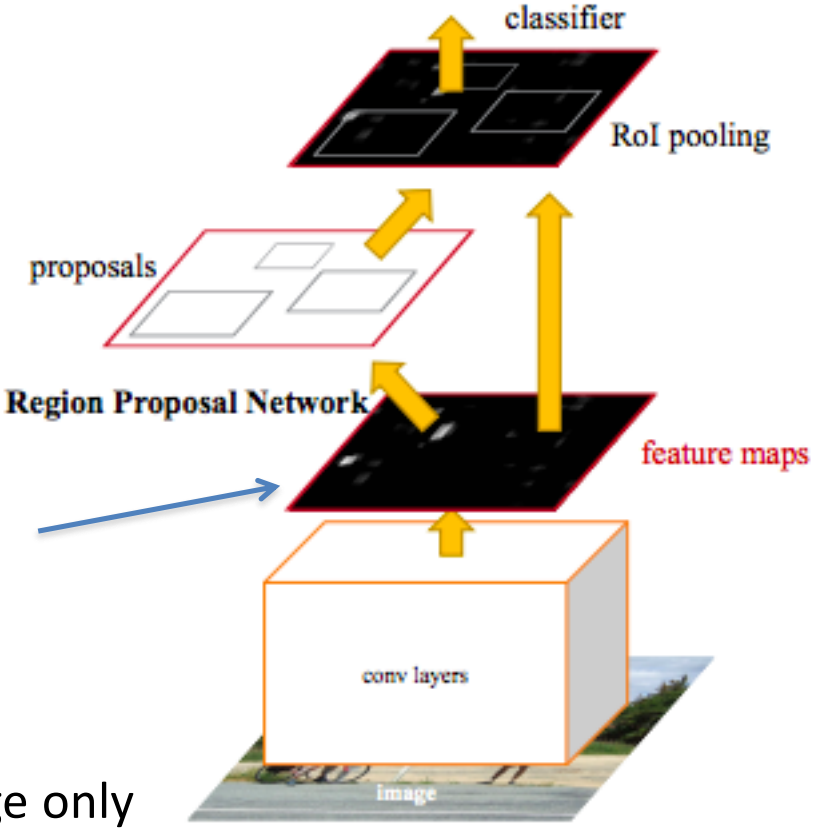
L_{loc}: basically SE between GT and prediction
Sum the SE for 4 parameters
smooth_{L1}(x): less sensitive to outliers

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

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

Faster R-CNN

- Selective search becomes the bottleneck
- Region proposal network



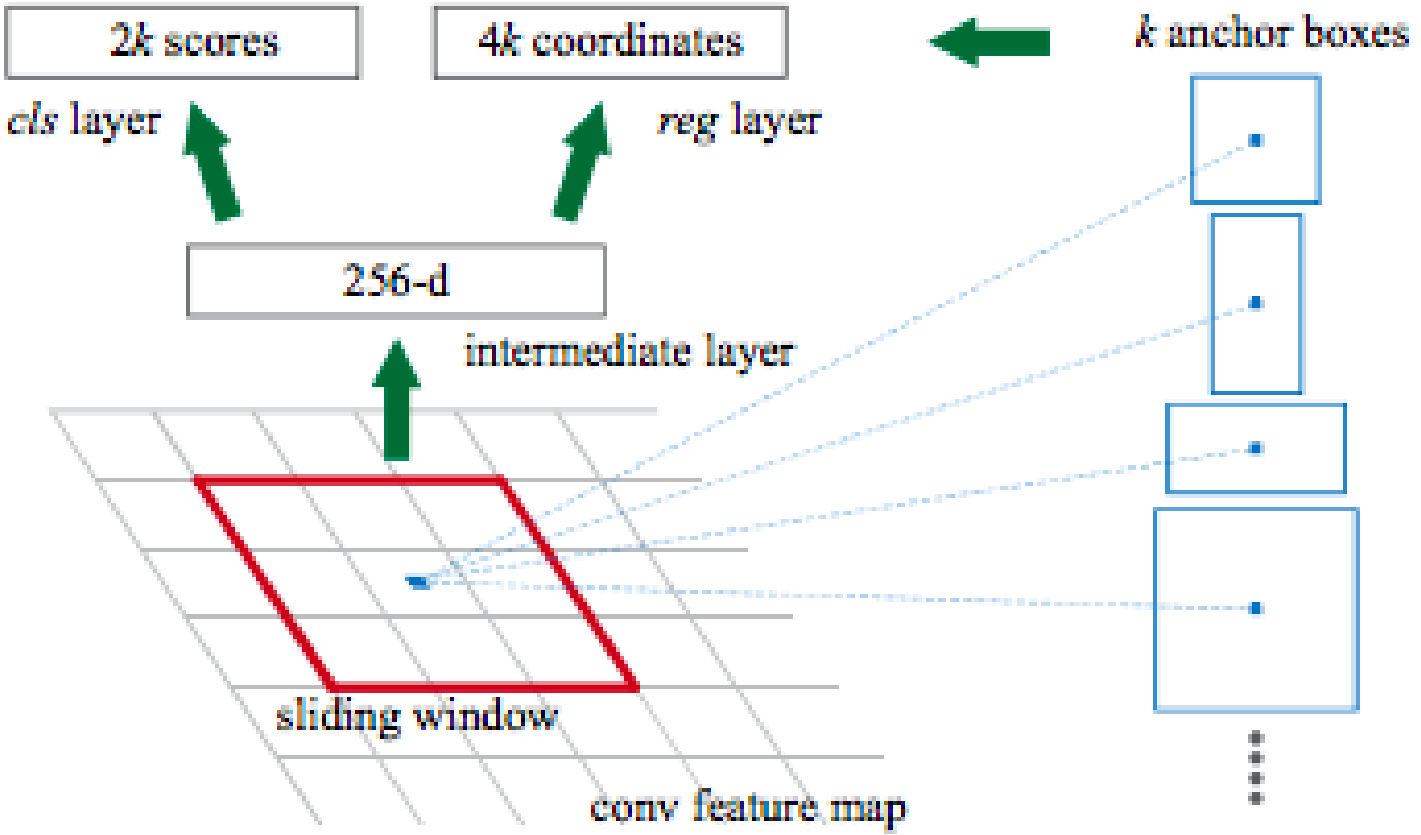
Feature maps used by region-based detectors can also be used for generating region proposals

Input: Image only

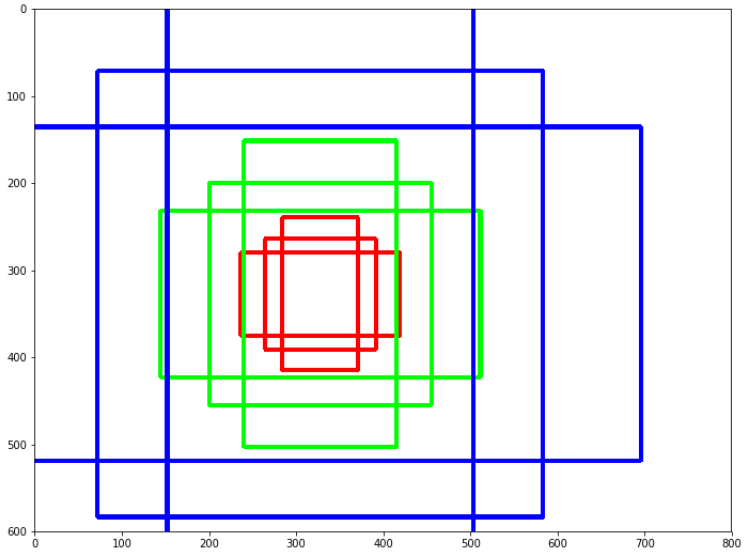
RPN

Slides a 3x3 window over the CNN feature

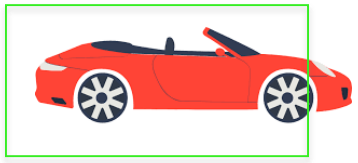
At each window location, RPN outputs a score and a bbox per anchor (total k anchors)



Anchor

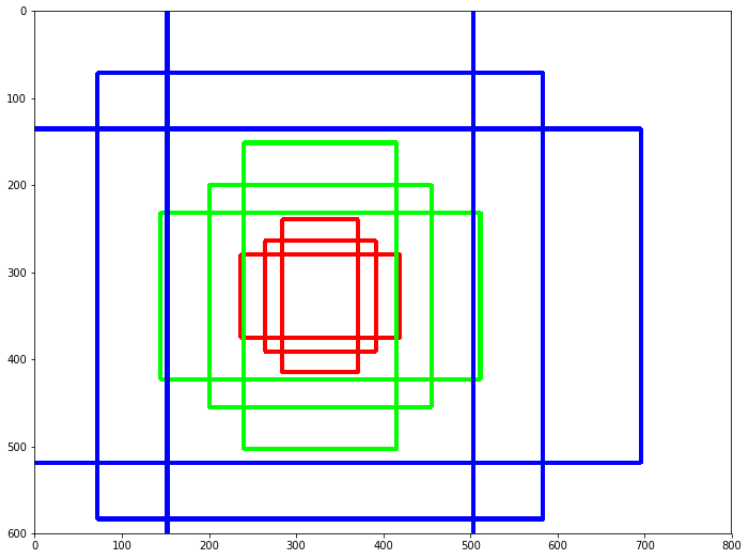


Common aspect ratio and size -> guide RPN



Positive label for this anchor (IOU overlap > 0.7)

Loss function



A mini-batch of positive/negative anchors (index i)

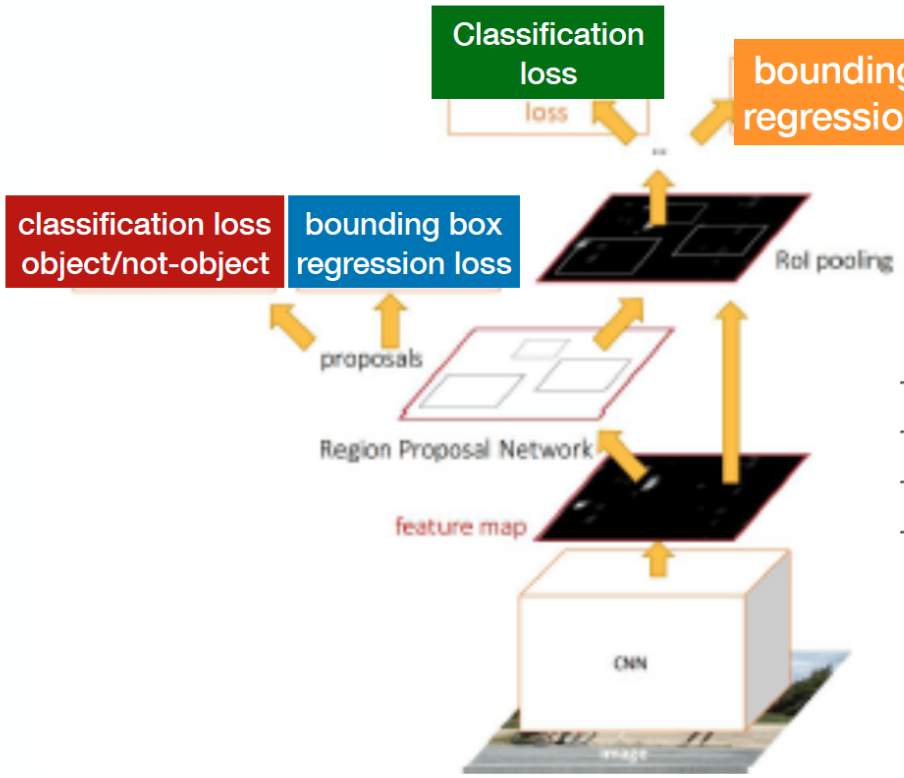
Ground-truth: 1 if this anchor overlaps with an object (positive), 0 otherwise

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

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

Robust loss function for bbox parameter

Faster R-CNN

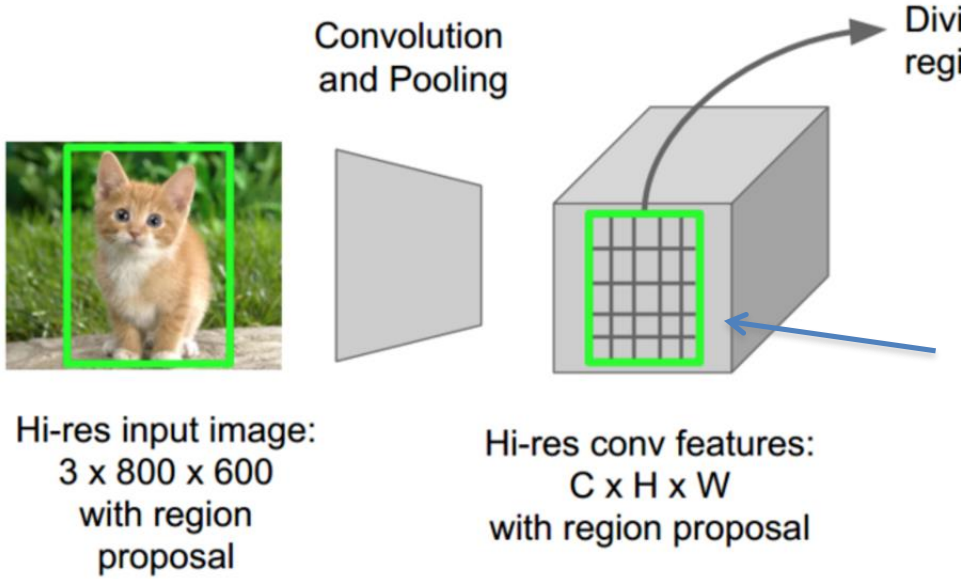


One network, 4 losses

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

backup

Back propagation through ROI Pooling



$$\frac{\partial L}{\partial x_i} = \sum_r \sum_j [i = i^*(r,j)] \frac{\partial L}{\partial y_{rj}}$$

Input to ROI pooling: x_i

For ROI r in the mini-batch and for each pooling output y_{rj}, partial derivate is accumulated if i is the argmax selected for y_{rj}