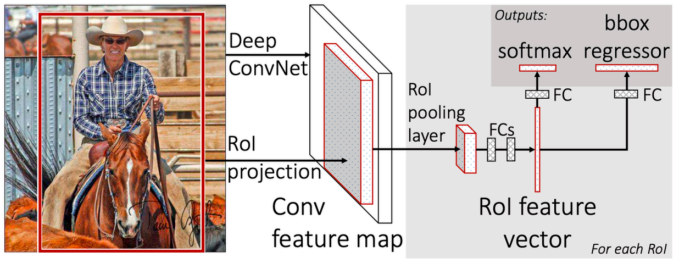
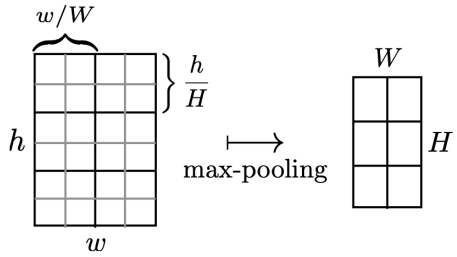
Fast R-CNN

YouTube Playlist



RoI pooling layer

$$(\underbrace{r, c}_{\text{top-left corner}}, \overbrace{h, w}^{\text{height \& width}}) \rightarrow \text{RoI (Regions of Interest)}$$



H & W are set to be compatible with the first fully connected layer

Fine-tuning

Mini-batches are sampled hierarchically, first by sampling N images and then by sampling R/N RoIs from each image (N=2, R = 128).

Multi-task loss

$p = (p_0, \ldots, p_K) \rightarrow$ probability distribution over $K+1$ categories

$t^k = (t^k_x, t^k_y, t^k_w, t^k_h) \rightarrow$ bounding box regression offsets

$k = 1, \ldots, K \rightarrow$ object classes

category-specific bounding-box regressors

Each training RoI is labeled with:

$u \rightarrow$ ground-truth class

$v \rightarrow$ ground-truth bounding box regression target

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

$$L_{\text{cls}}(p, u) = -\log p_u$$

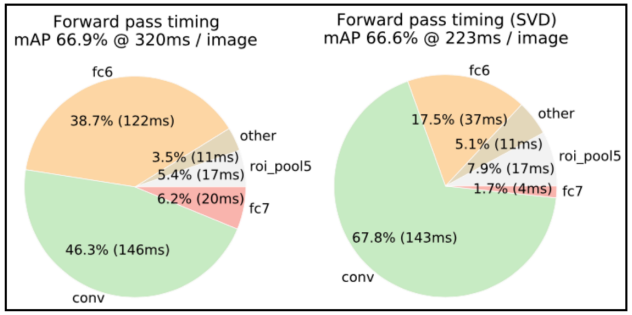
$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{\text{x,y,w,h}\}} \text{smooth}_{L_1}(t^u_i - v_i)$$

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

robust L_1 loss that is less sensitive to outliers

Truncated SVD

$$W \approx U \Sigma_t V^T$$



	Fast R-CNN			R-CNN			SPPnet
	S	M	L	S	M	L	†L
train time (h)	**1.2**	2.0	9.5	22	28	84	25
train speedup	**18.3**×	14.0×	8.8×	1×	1×	1×	3.4×
test rate (s/im)	0.10	0.15	0.32	9.8	12.1	47.0	2.3
▷ with SVD	**0.06**	0.08	0.22	-	-	-	-
test speedup	98×	80×	146×	1×	1×	1×	20×
▷ with SVD	169×	150×	**213**×	-	-	-	-
VOC07 mAP	57.1	59.2	**66.9**	58.5	60.2	66.0	63.1
▷ with SVD	56.5	58.7	66.6	-	-	-	-

Back-propagation through RoI pooling layers

$x_i \in \mathbb{R} \rightarrow i$-th activation input into the RoI pooling layer

$y_{rj} \in \mathbb{R} \rightarrow$ layer's j-th output from the r-th RoI

$\mathcal{R}(r, j) \rightarrow$ sub-window over which the output unit y_{rj} max-pools

$$i^*(r, j) = \arg\max_{i' \in \mathcal{R}(r,j)} x_{i'}$$

$$y_{rj} = x_{i^*(r,j)}$$

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

A single x_i may be assigned to several different outputs y_{rj}.

Girshick, Ross. "Fast r-cnn." Proceedings of the IEEE international conference on computer vision. 2015.