Fast R-CNN

이재형

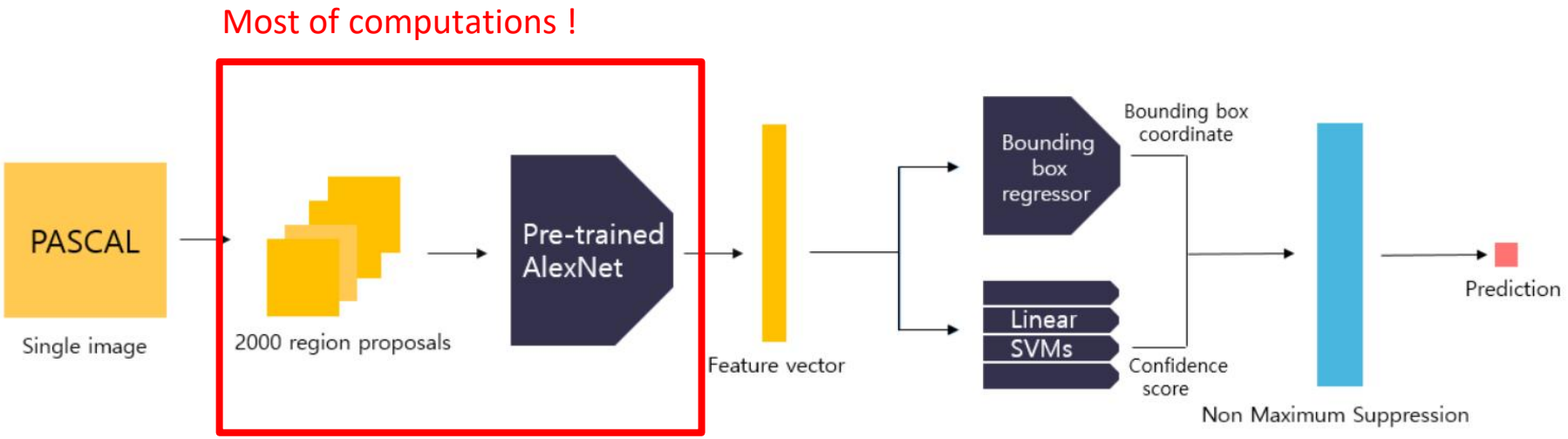
Signal Processing & Artificial-intelligence Lab

Hanyang University

Contents

Drawbacks of R-CNN

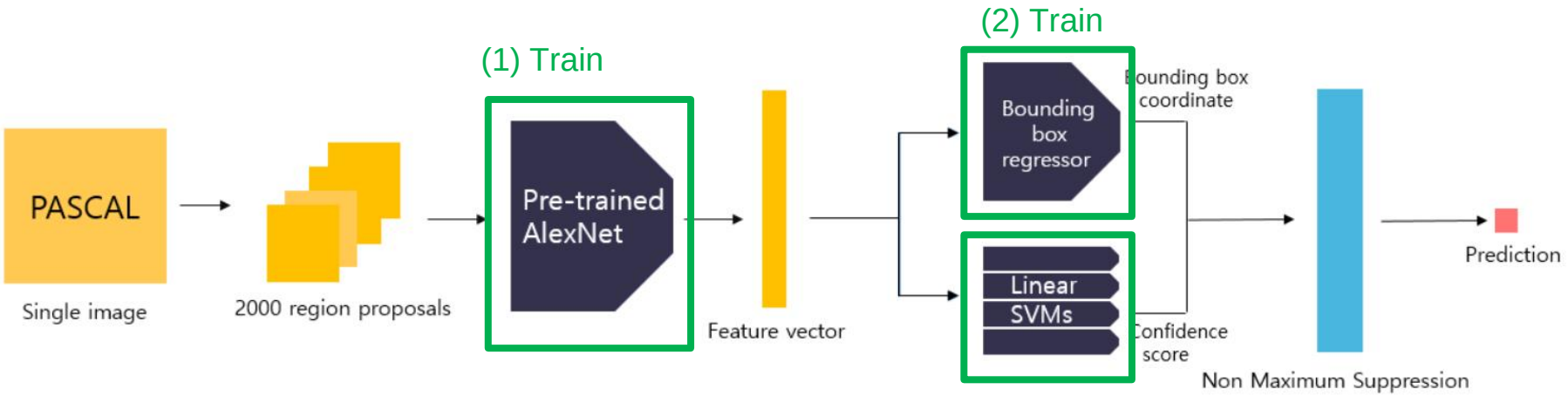
1. Very **slow inference** speed
   - 1 image inference time = 47sec
2. **Expansive training** cost in space and time
   - Need to save AlexNet output feature = hundreds of GB !
3. Separated training pipeline : **complex**
   - AlexNet, Linear SVM, bbox regressor



R-CNN Architecture

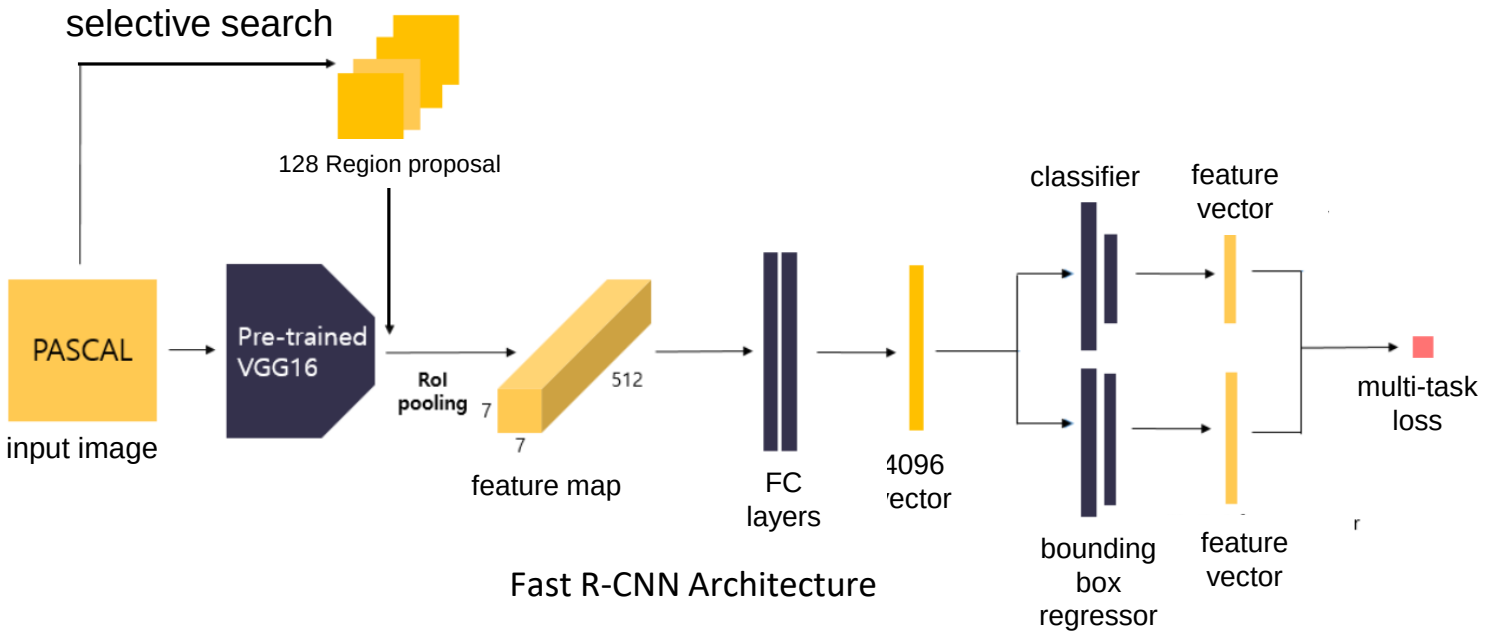
Drawbacks of R-CNN

1. Very **slow inference** speed
   - 1 image inference time = 47sec
2. **Expansive training** cost in space and time
   - Need to save AlexNet output feature = hundreds of GB !
3. Separated training pipeline : **complex**
   - AlexNet, Linear SVM, bbox regressor



R-CNN Architecture

Fast R-CNN – Architecture

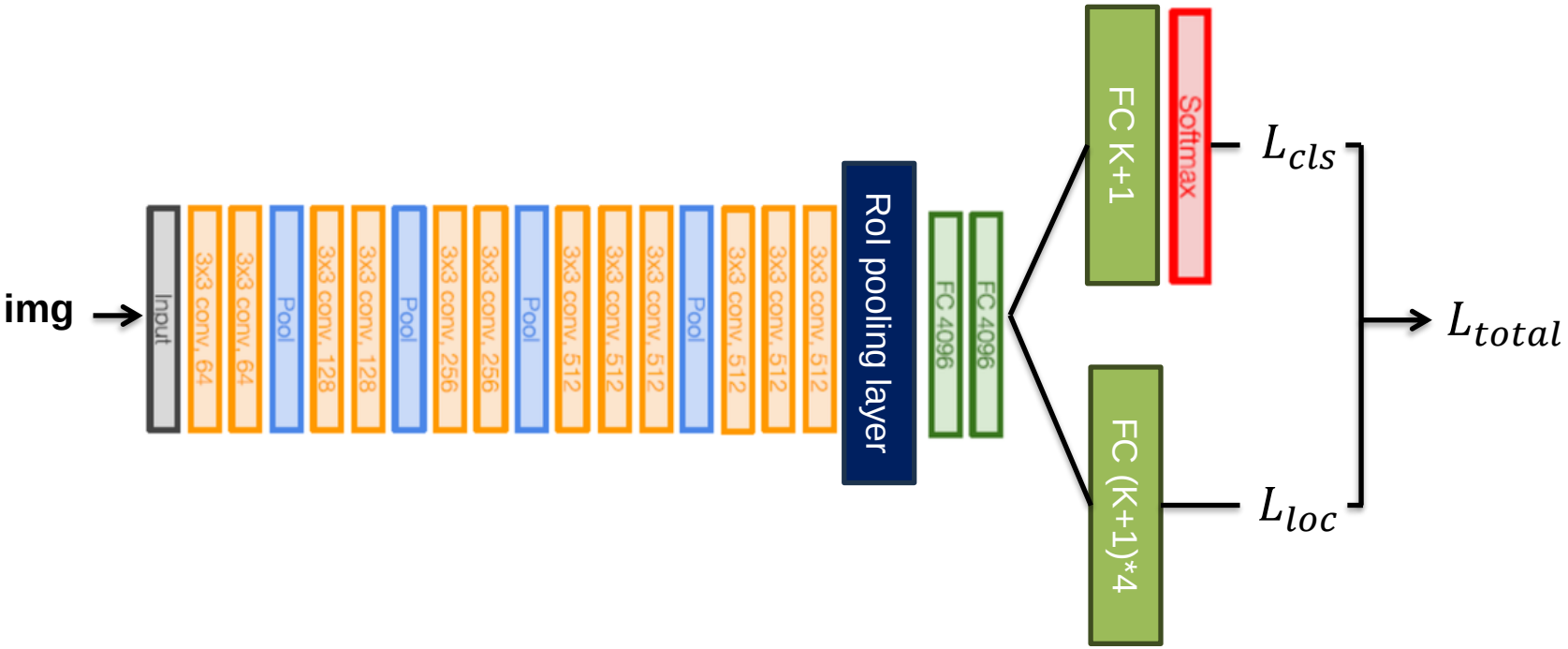


Fast R-CNN Architecture

Fast R-CNN – Architecture

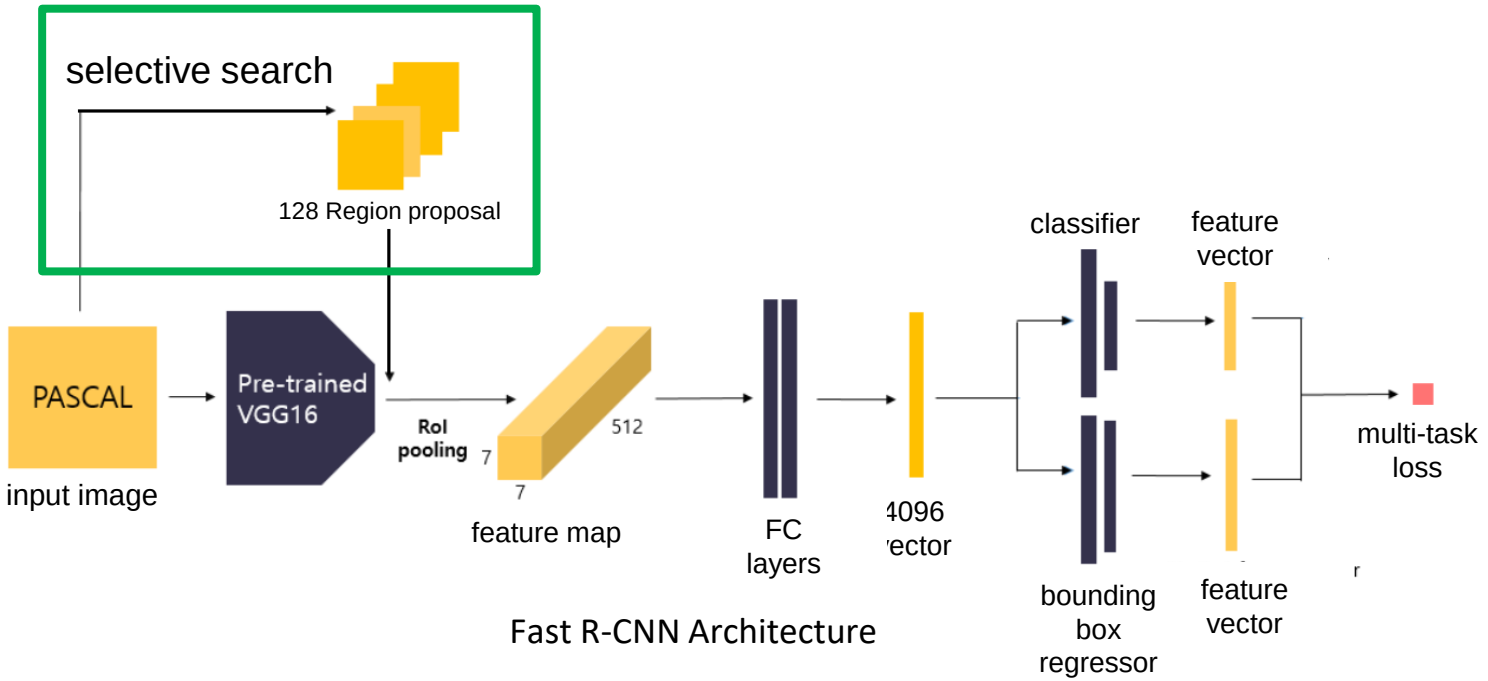
VGG-16

Fast R-CNN – Architecture



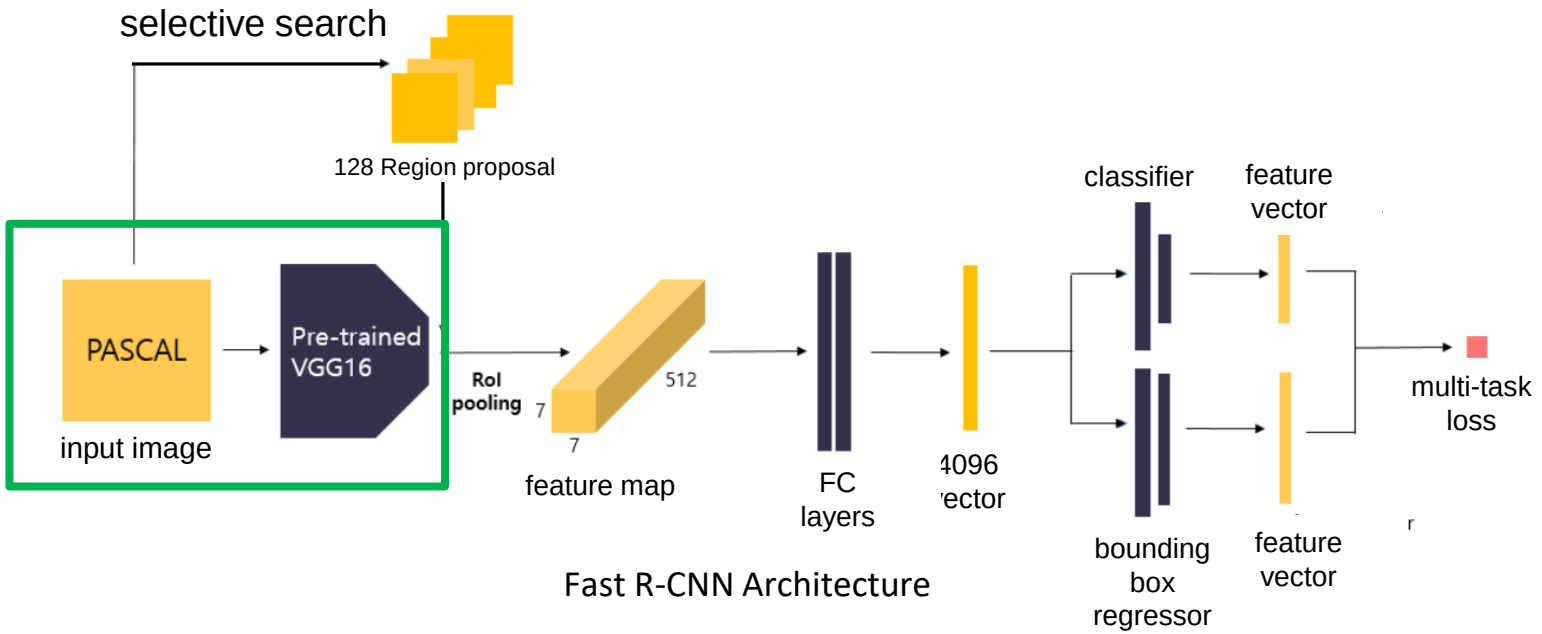
"Fast R-CNN"

Fast R-CNN – Training



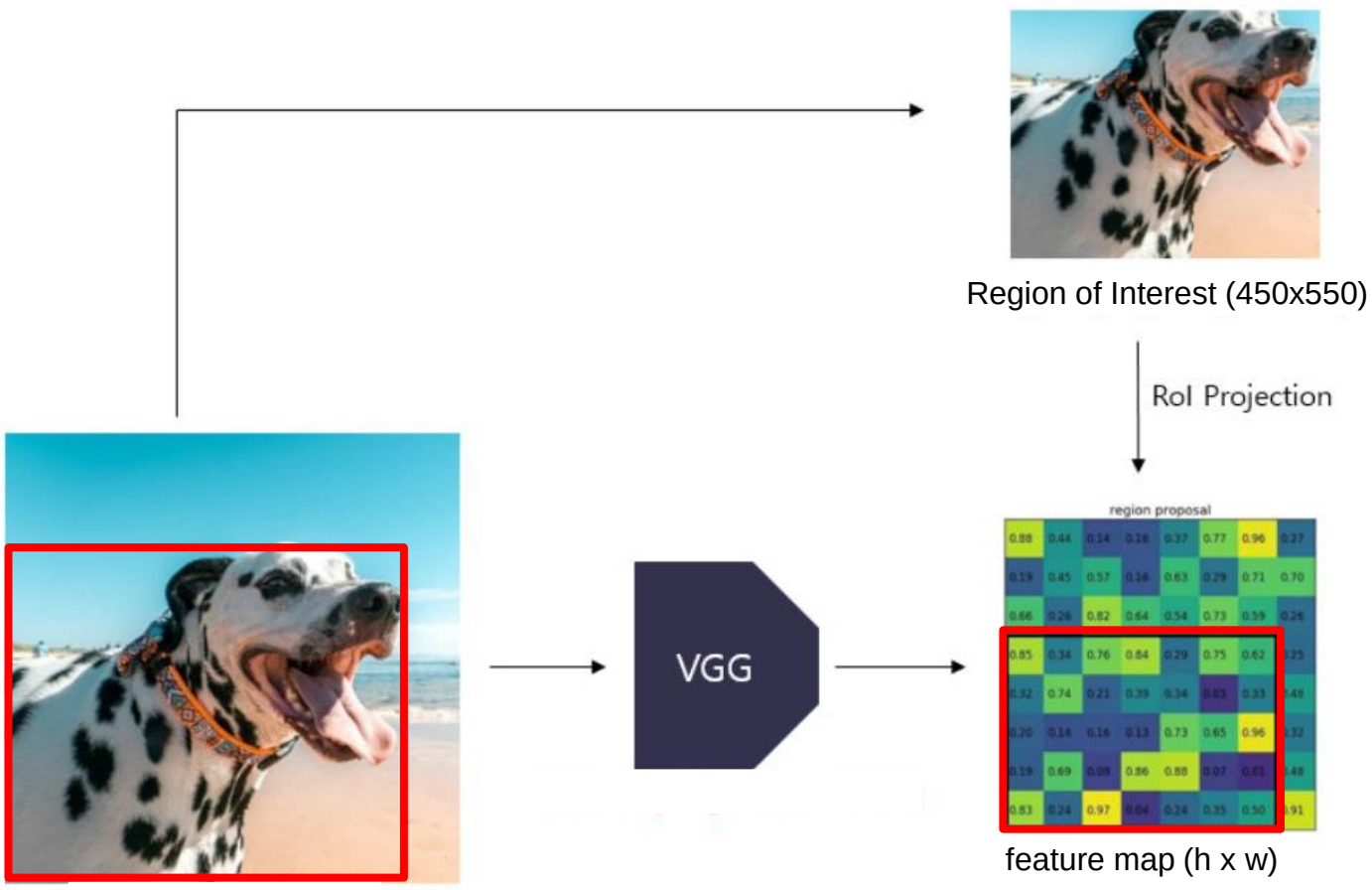
Fast R-CNN Architecture

Fast R-CNN – Training



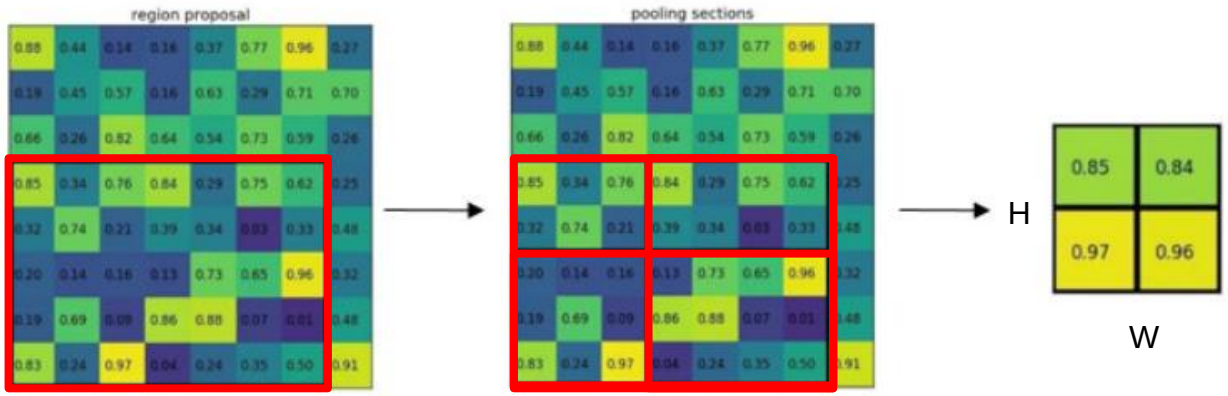
Fast R-CNN Architecture

Fast R-CNN – Training



RoI projection

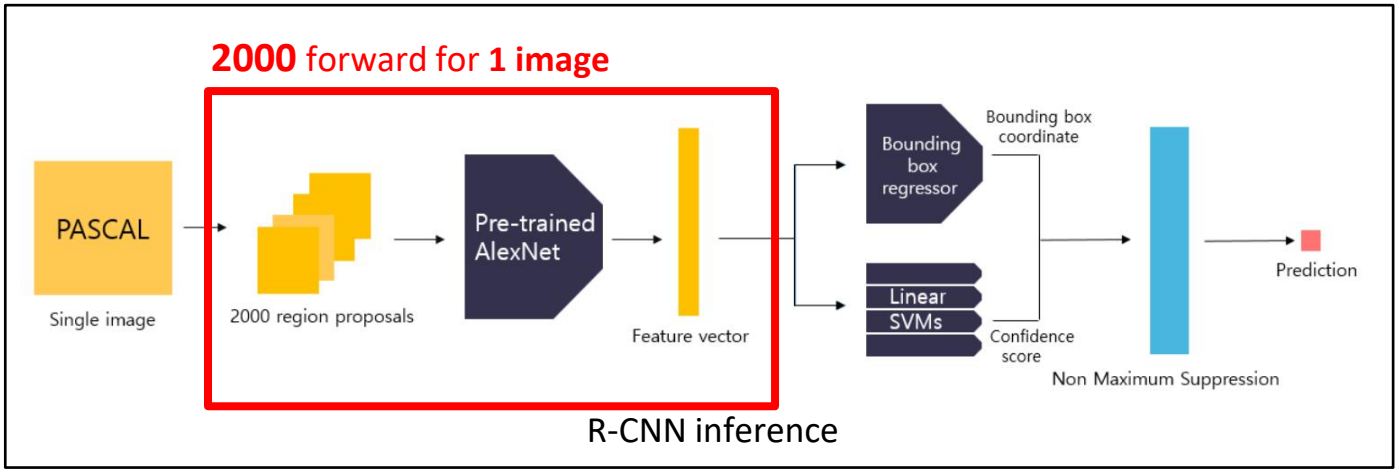
Fast R-CNN – Training

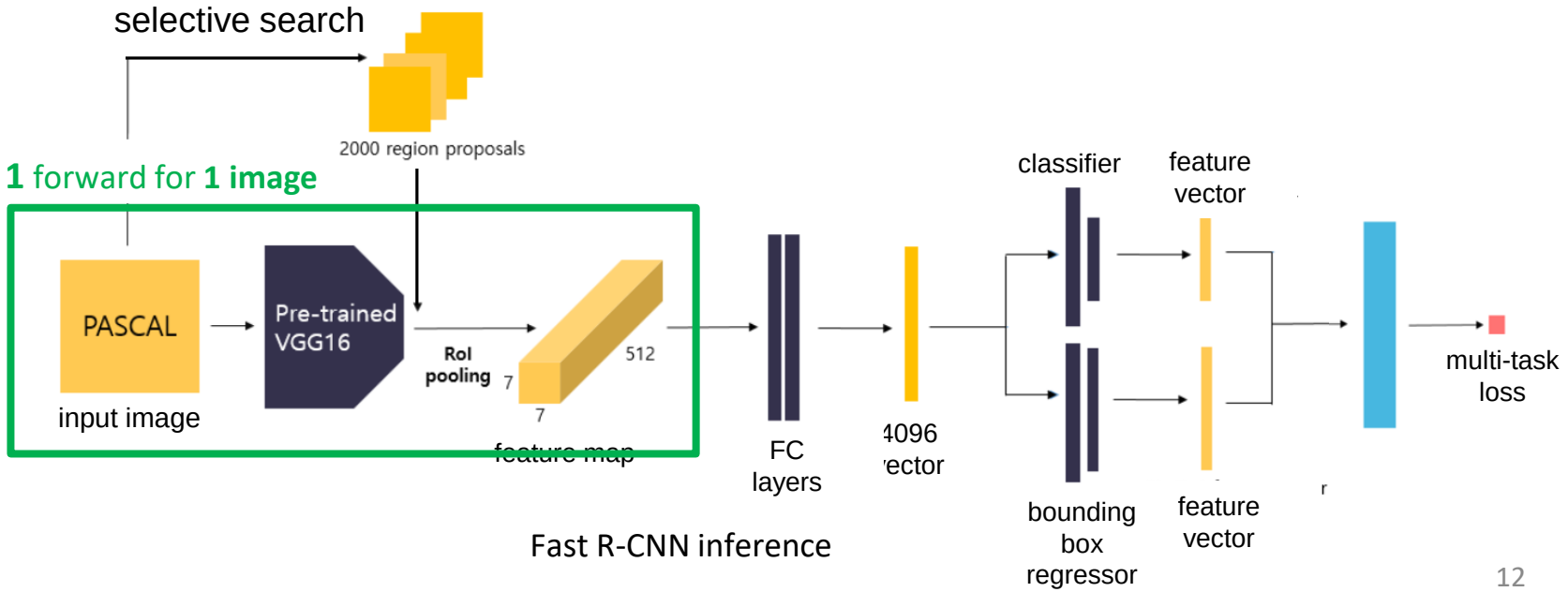


feature map (h x w)

RoI pooling

Fast R-CNN – feature sharing



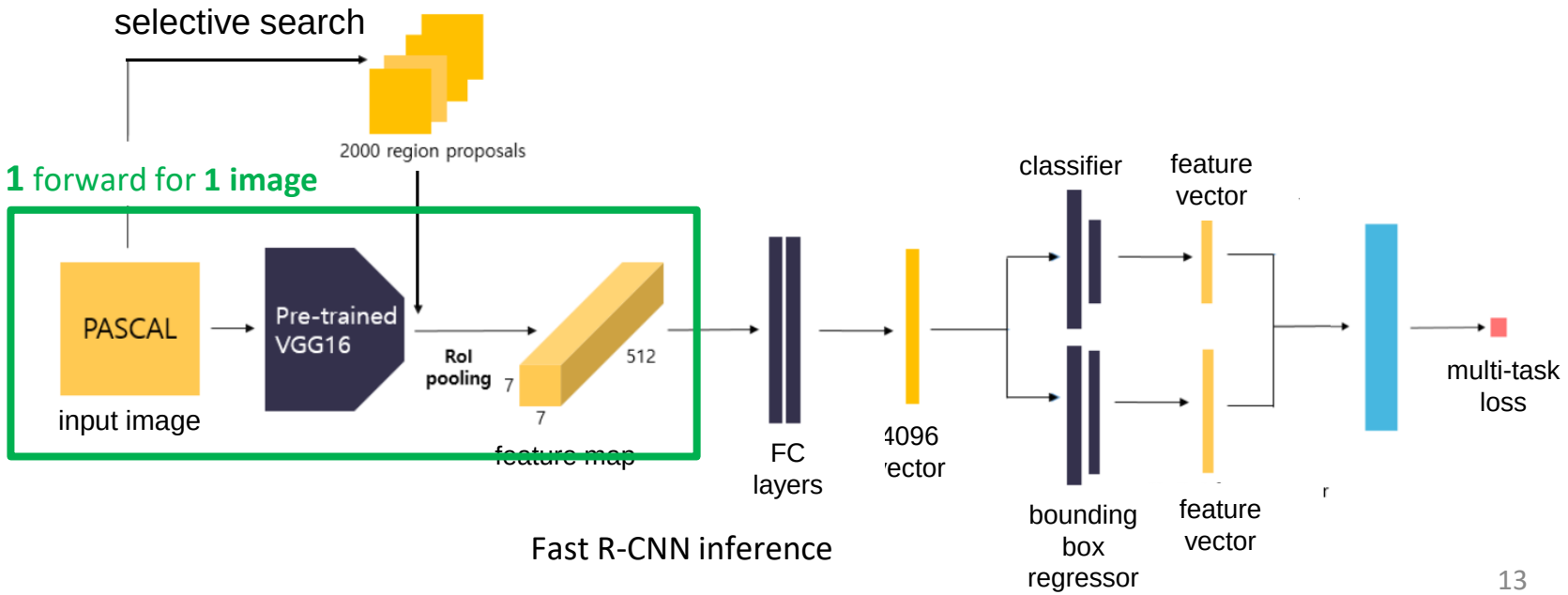


Fast R-CNN – Training

much faster

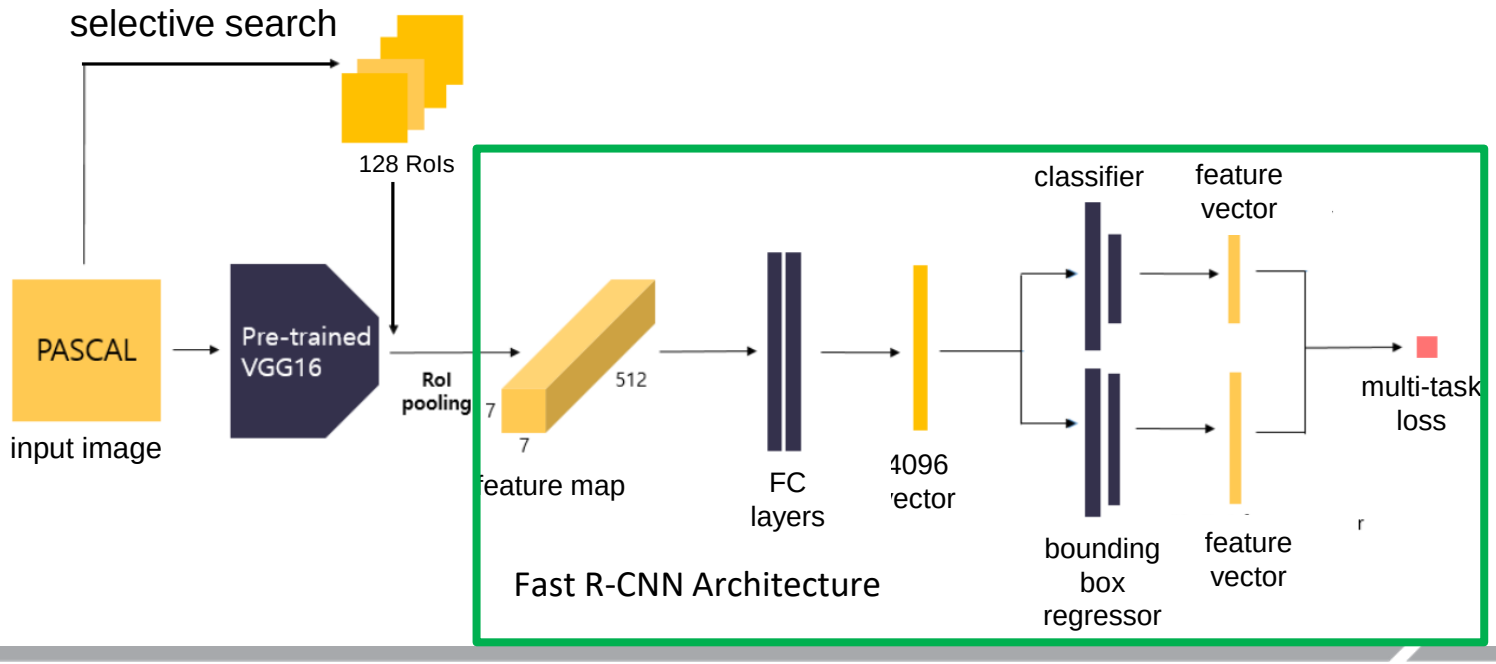
	Fast R-CNN			R-CNN		
	S	**M**	**L**	**S**	**M**	**L**
train time (h)	**1.2**	2.0	9.5	22	28	84
train speedup	**18.3**×	14.0×	8.8×	1×	1×	1×
test rate (s/im)	0.10	0.15	0.32	9.8	12.1	47.0
▷ with SVD	**0.06**	0.08	0.22	-	-	-
test speedup	98×	80×	146×	1×	1×	1×
▷ with SVD	169×	150×	**213**×	-	-	-
VOC07 mAP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

S : AlexNet
M : AlexNet (Wider)
L : VGG-16



Fast R-CNN inference

Fast R-CNN – Training

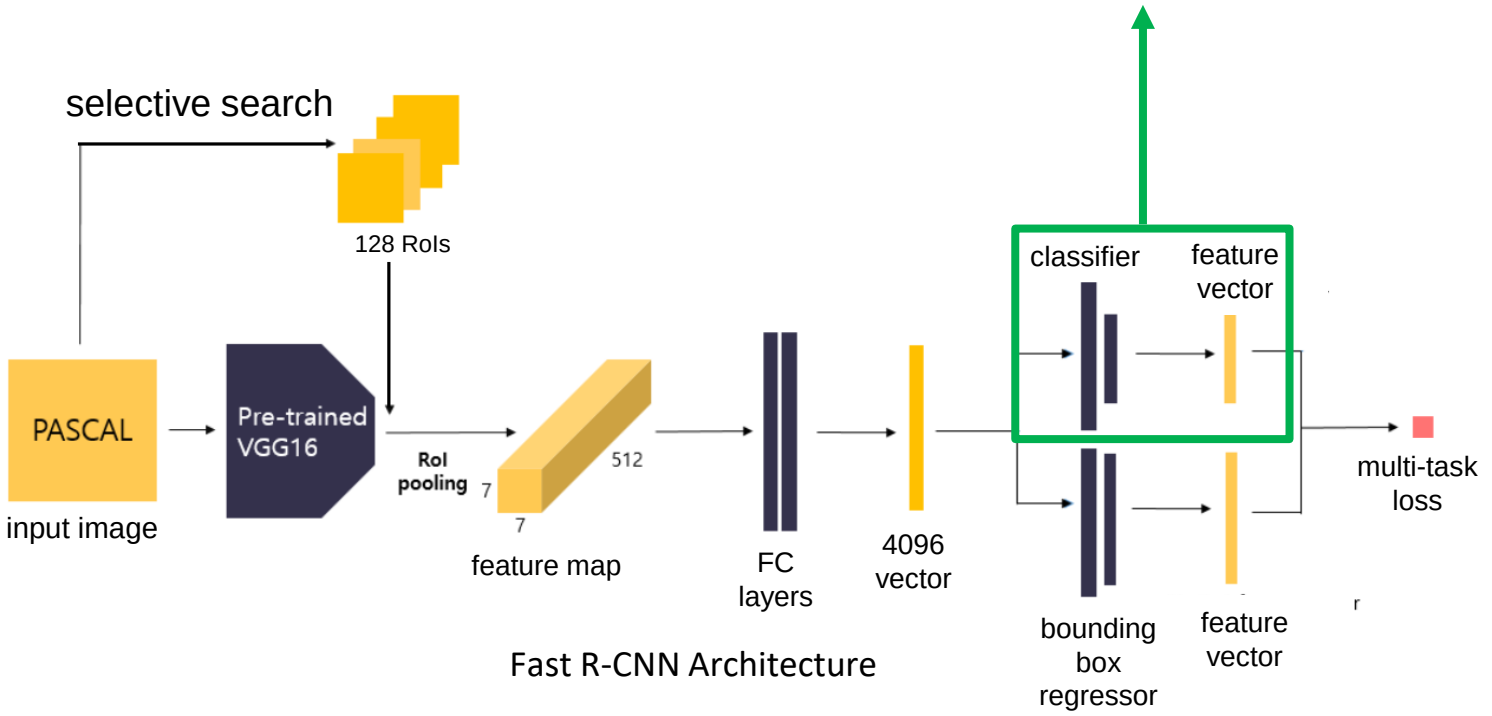


Fast R-CNN – Training

Input 4096 size feature vector
output (K+1) sized vector

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

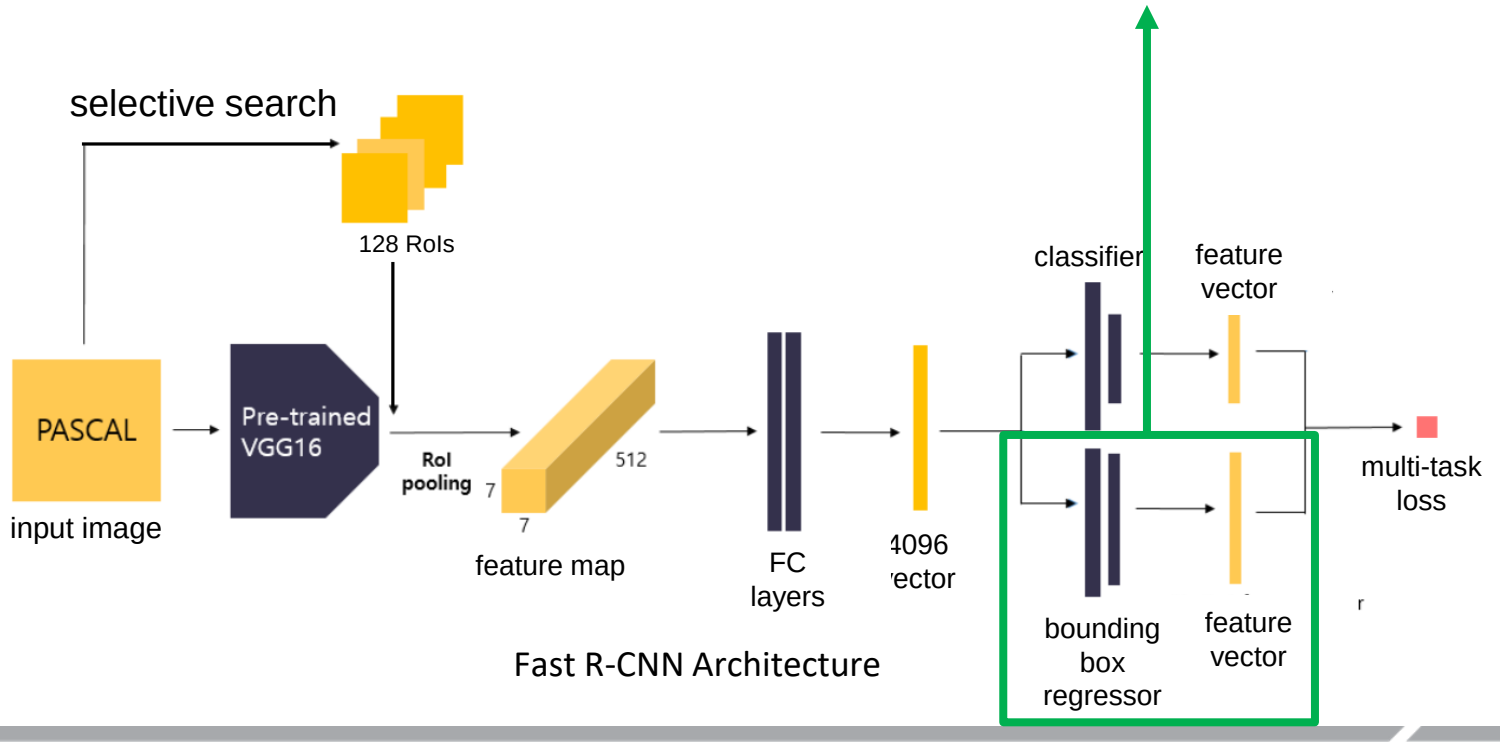
$\boldsymbol{p}$ **:** output probability distribution
$\boldsymbol{u}$: ground truth class



Fast R-CNN Architecture

Fast R-CNN – Training

Input 4096 size feature vector
output 4*(K+1) sized vector

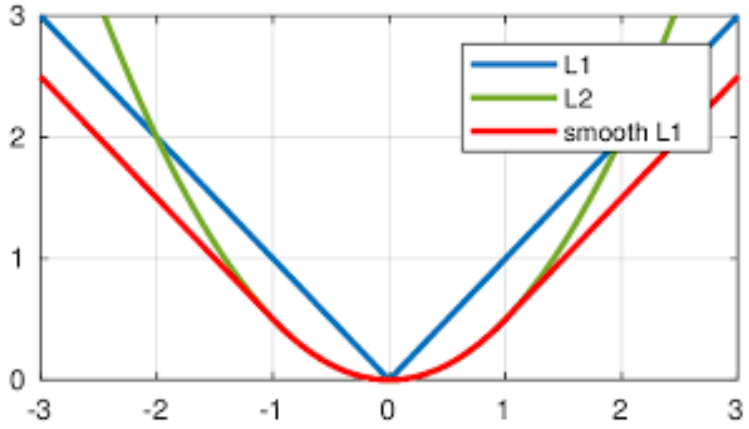


Fast R-CNN Architecture

Fast R-CNN – Training

bounding box regression Loss

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



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

pred location $t^k = (t_{\text{x}}^k, t_{\text{y}}^k, t_{\text{w}}^k, t_{\text{h}}^k)$

ground truth $v = (v_{\text{x}}, v_{\text{y}}, v_{\text{w}}, v_{\text{h}})$

Fast R-CNN – Training

Multi-task Loss

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

for background class(u=0), ignore L_loc

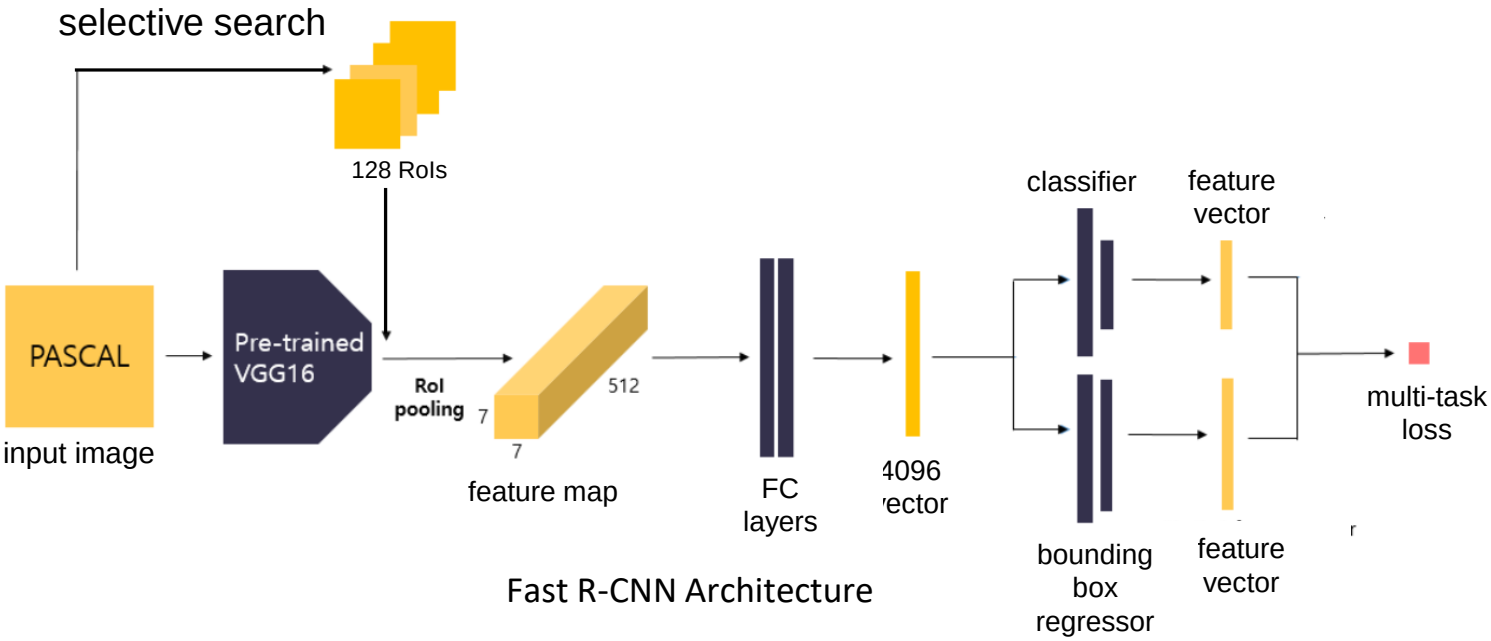
u : ground truth class

	S				M				L			
multi-task training?		✓		✓		✓		✓		✓		✓
stage-wise training?			✓				✓				✓	
test-time bbox reg?			✓	✓			✓	✓			✓	✓
VOC07 mAP	52.2	53.3	54.6	**57.1**	54.7	55.5	56.6	**59.2**	62.6	63.4	64.0	**66.9**

S : AlexNet
M : AlexNet (Wider)
L : VGG-16

multi-task learning is good for this model !

Conclusion



Fast R-CNN Architecture

Thank You

이재형

Signal Processing & Artificial-intelligence Lab

Hanyang University