TensorFlow Object Detection API에서 제공하는 다양한 Object Detection을 위한 최신 모델들

- TensorFlow Object Detection API는 다음과 같은 최신 Object Detection 모델의 다양한 backbone을 이용한 구현을 제공합니다.

① Faster R-CNN

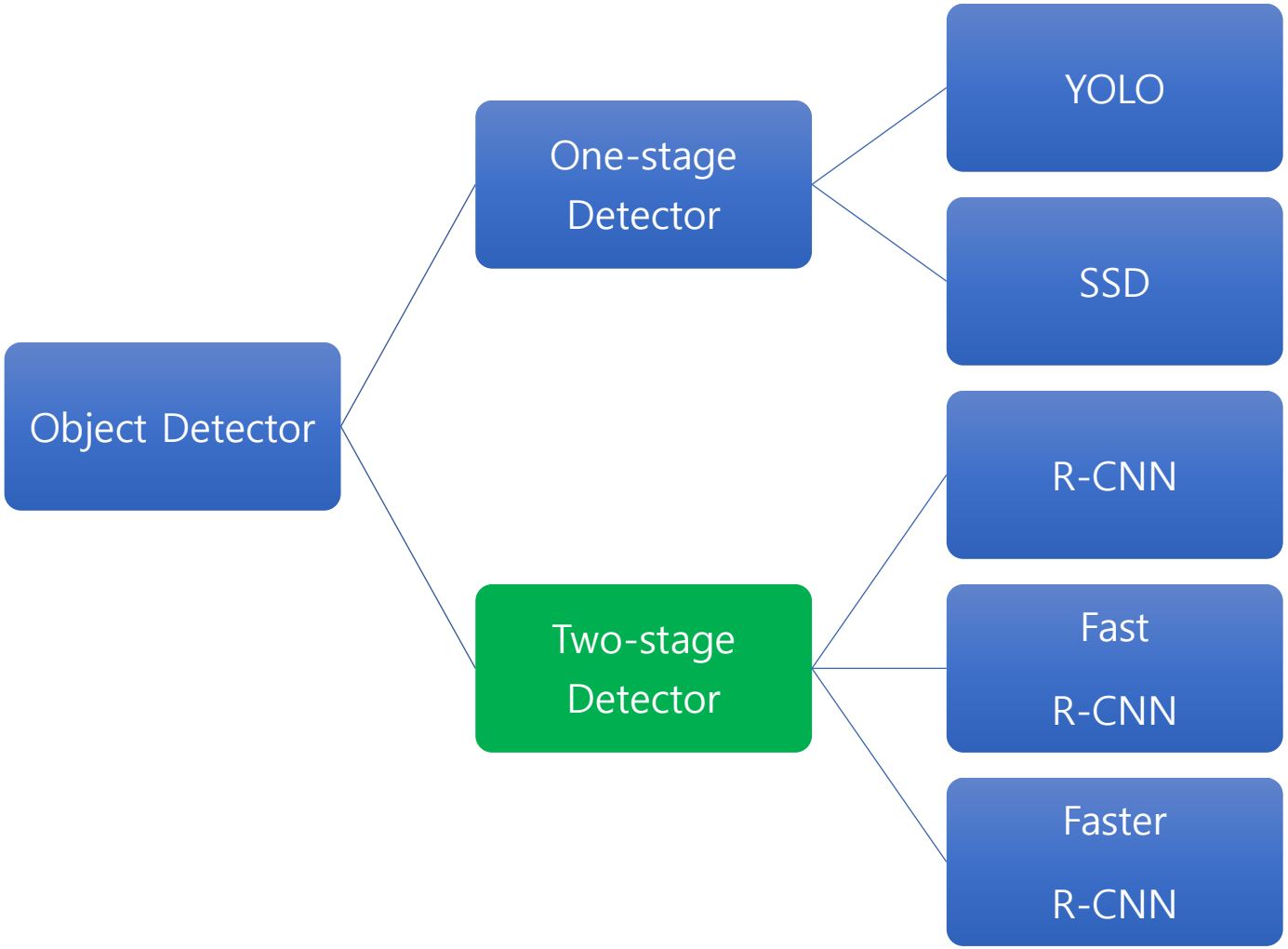
② SSD(Single Shot Multi-box Detector)

③ RetinaNet

④ CenterNet

⑤ EfficientDet

One-stage Detector vs Two-Stage Detector



Faster R-CNN 발전과정

Fast R-CNN

- Girshick, Ross. "Fast r-cnn." Proceedings of the IEEE international conference on computer vision. 2015.
- https://arxiv.org/pdf/1504.08083.pdf

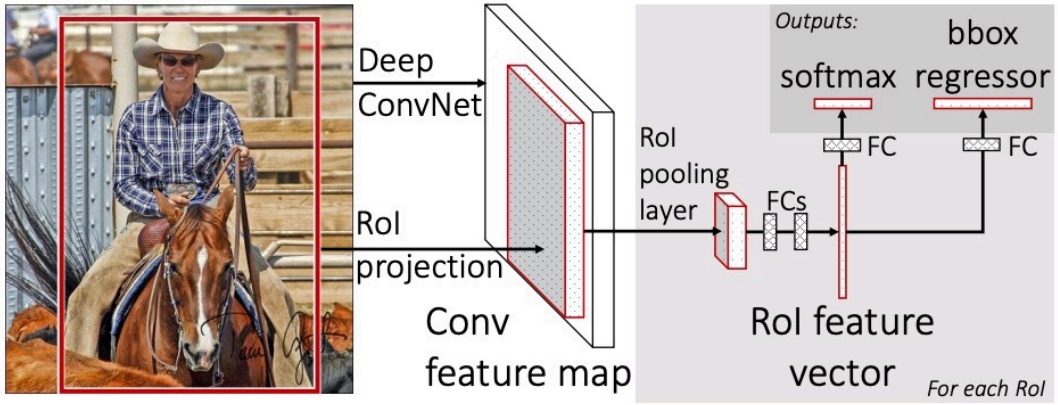
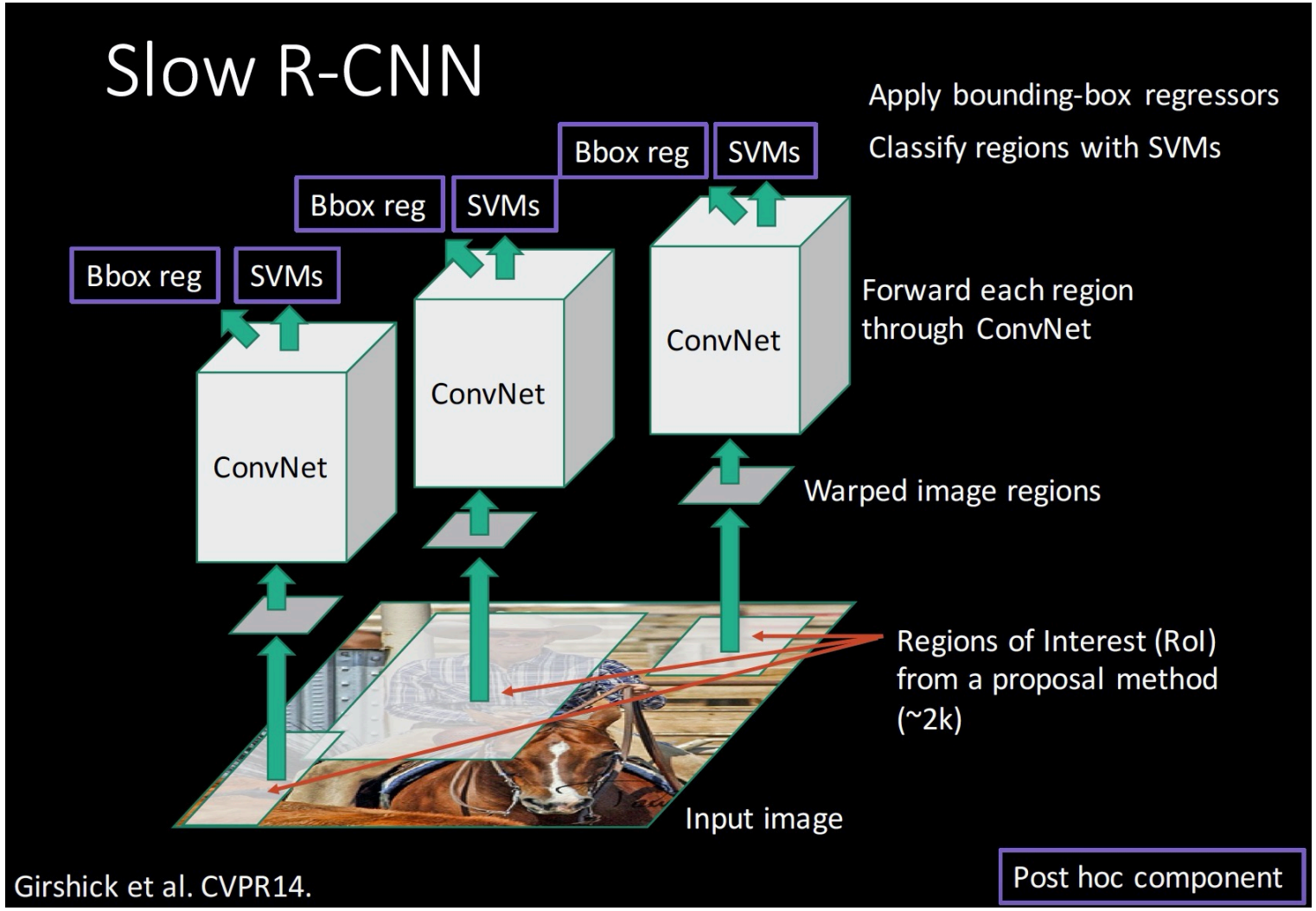


Figure 1. Fast R-CNN architecture. An input image and multiple regions of interest (RoIs) are input into a fully convolutional network. Each RoI is pooled into a fixed-size feature map and then mapped to a feature vector by fully connected layers (FCs). The network has two output vectors per RoI: softmax probabilities and per-class bounding-box regression offsets. The architecture is trained end-to-end with a multi-task loss.

R-CNN Architecture의 문제점

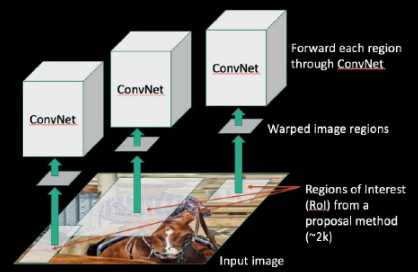


Slide Credit : Ross Girshick
https://www.rossgirshick.info/

R-CNN Architecture의 문제점

What's wrong with slow R-CNN?

- Ad hoc training objectives
  - Fine-tune network with softmax classifier (log loss)
  - Train post-hoc linear SVMs (hinge loss)
  - Train post-hoc bounding-box regressions (least squares)
- Training is slow (84h), takes a lot of disk space
- Inference (detection) is slow
  - 47s / image with VGG16 [Simonyan & Zisserman. ICLR15]
  - Fixed by SPP-net [He et al. ECCV14]

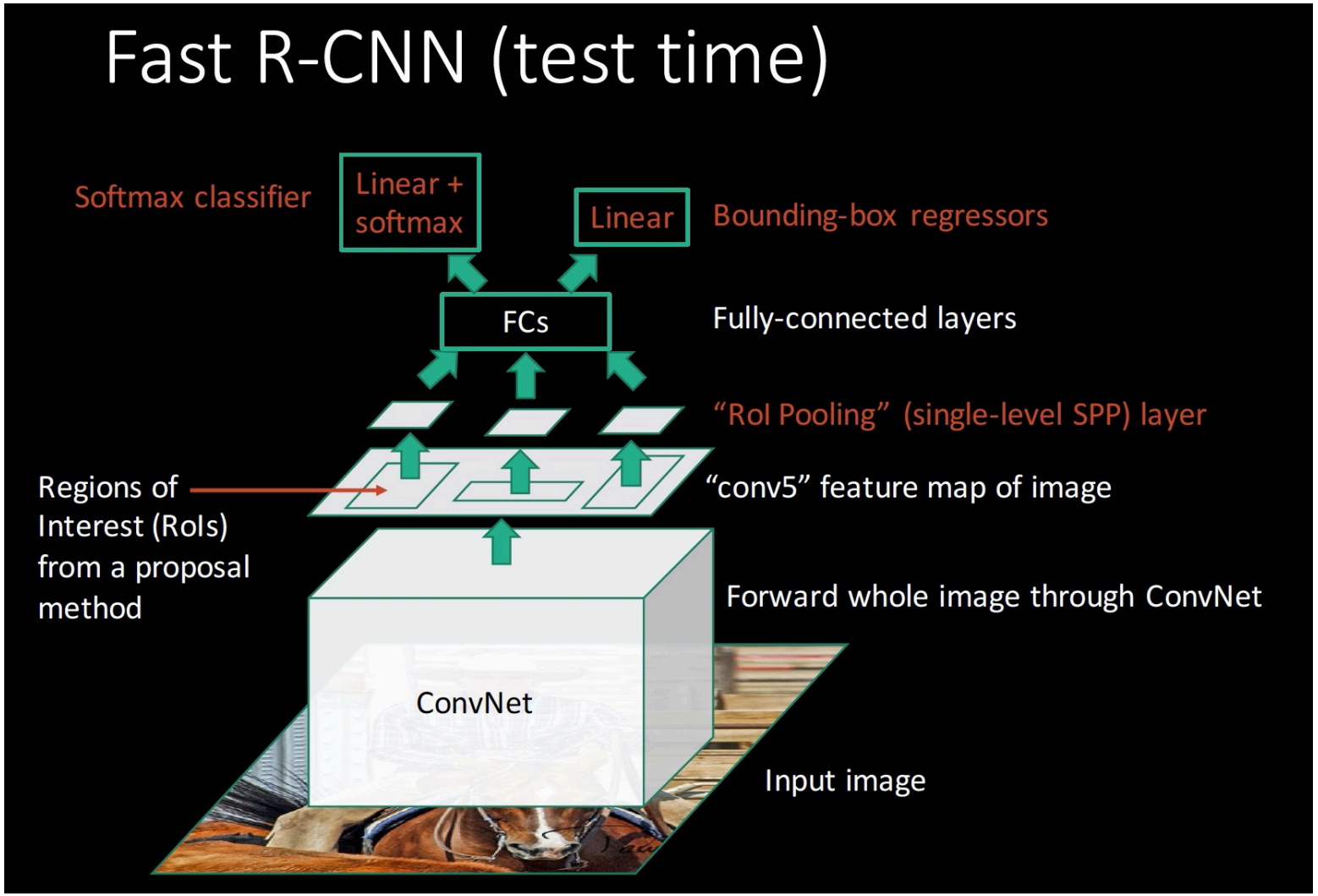


~2000 ConvNet forward passes per image

Slide Credit : Ross Girshick
https://www.rossgirshick.info/

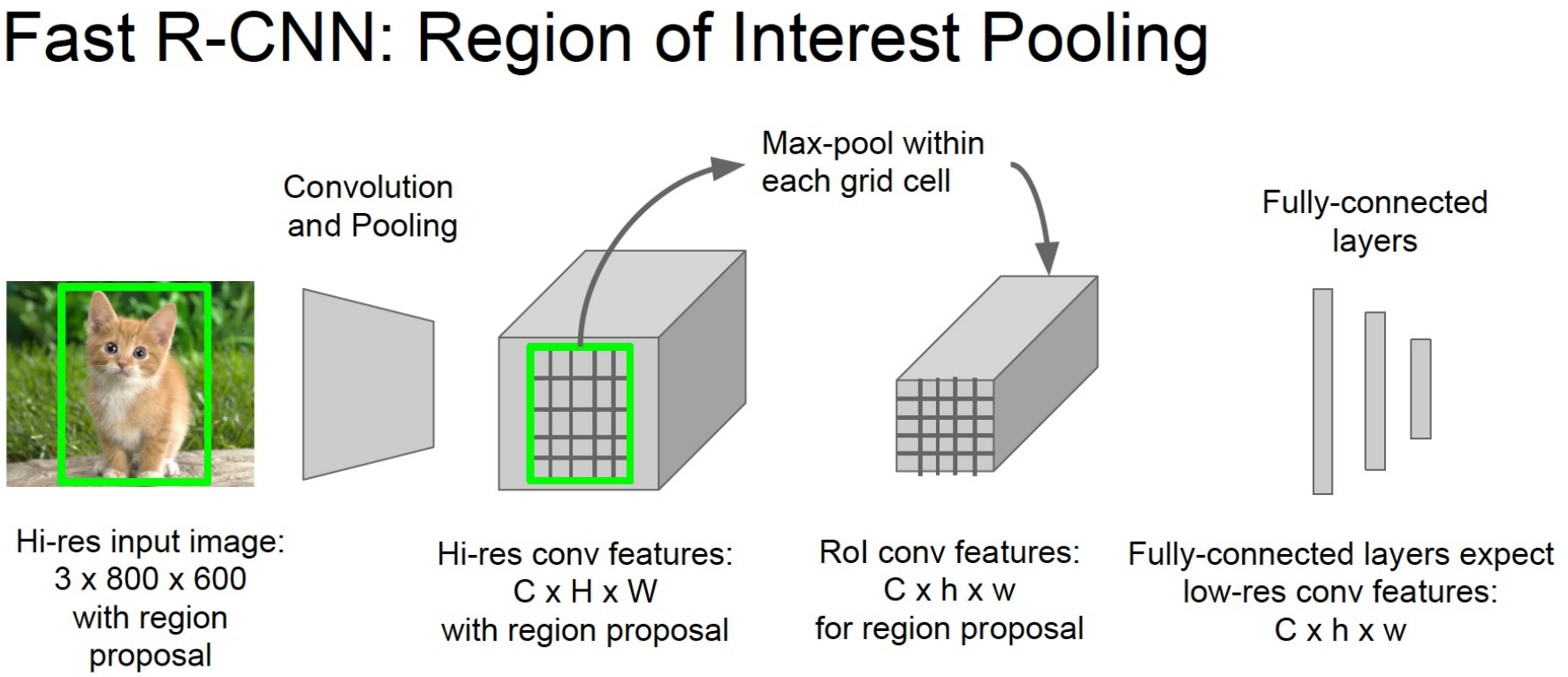
AI School.

Fast R-CNN Architecture



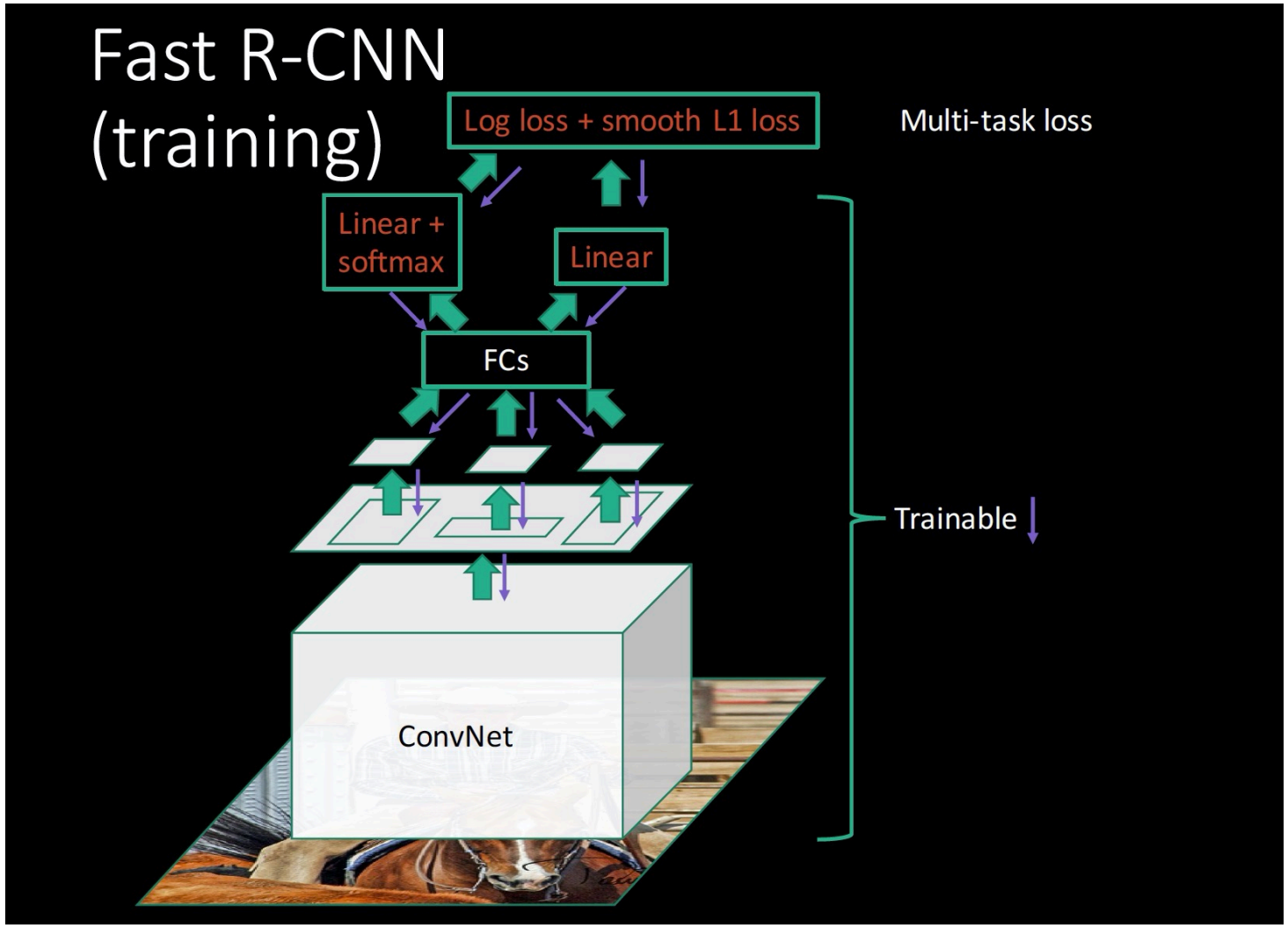
Slide Credit : Ross Girshick
https://www.rossgirshick.info/

ROI(Region of Interest Pooling)



Slide credit: CS231n (https://goo.gl/QSA36t)

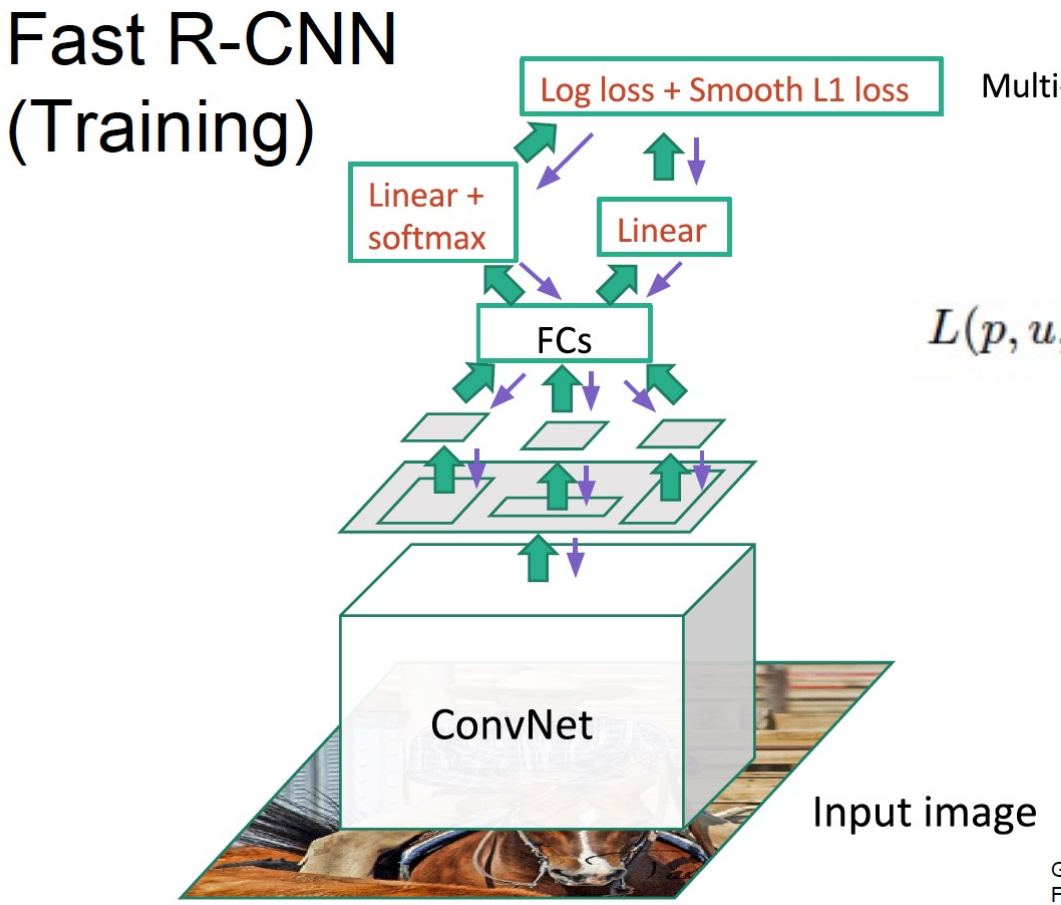
Fast R-CNN Architecture



Slide Credit : Ross Girshick
https://www.rossgirshick.info/

Fast R-CNN Multi-Task Loss

- All experiments use $\lambda = 1$.



Multi-Task Loss

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

Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Slide credit: CS231n (https://goo.gl/QSA36t)

Fast R-CNN Training Mini-batch

- During fine-tuning, each SGD mini-batch is constructed from **N = 2** images, chosen uniformly at random.
- We use mini-batches of size **R = 128**, sampling **64 RoIs from each image**.
- As in [9], we take **25% of the RoIs** from object proposals that have intersection over union **(IoU) overlap with groundtruth bounding box of at least 0.5**. These RoIs comprise the examples labeled with a **foreground object class**, i.e. $u \geq 1$.
- The **remaining RoIs** are sampled from object proposals that have a maximum **IoU with ground truth in the interval [0.1, 0.5)**, following [11]. These are the **background examples** and are labeled with $u = 0$.

Fast R-CNN Result

Main results

	Fast R-CNN	R-CNN [1]	SPP-net [2]
Train time (h)	**9.5**	84	25
- Speedup	**8.8x**	1x	3.4x
Test time / image	**0.32s**	47.0s	2.3s
Test speedup	**146x**	1x	20x
mAP	**66.9%**	66.0%	63.1%

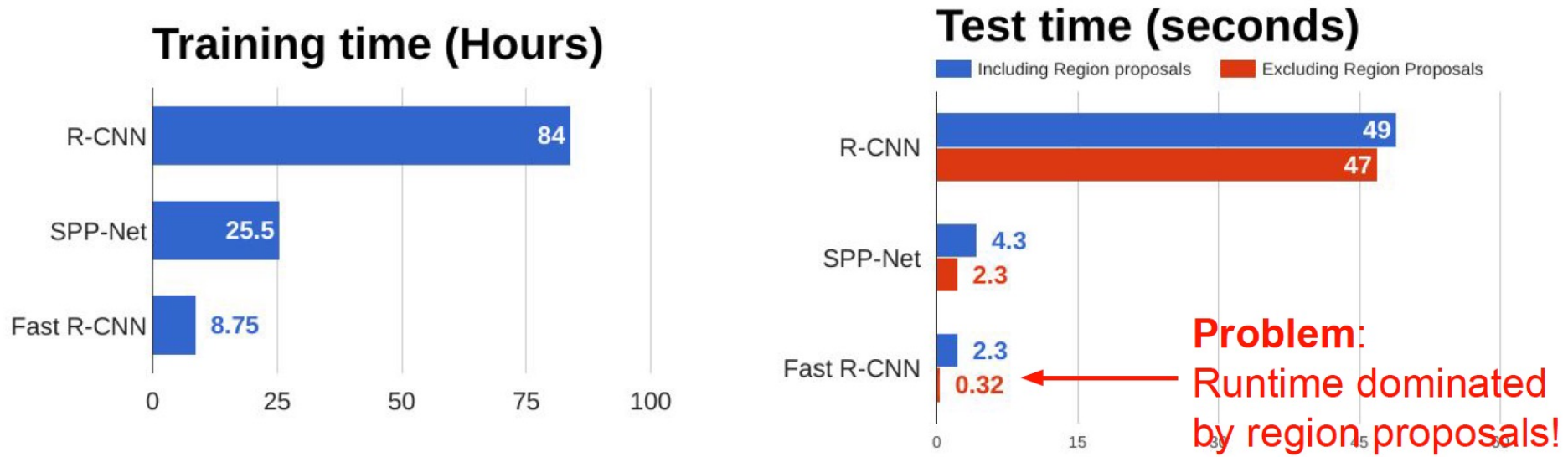
Timings exclude object proposal time, which is equal for all methods.
All methods use VGG16 from Simonyan and Zisserman.

[1] Girshick et al. CVPR14.
[2] He et al. ECCV14.

Slide Credit : Ross Girshick
https://www.rossgirshick.info/

Fast R-CNN Problem

R-CNN vs SPP vs Fast R-CNN



Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
He et al, "Spatial pyramid pooling in deep convolutional networks for visual recognition", ECCV 2014
Girshick, "Fast R-CNN", ICCV 2015

Slide credit: CS231n
(https://goo.gl/QSA36t)

Fast R-CNN의 장점과 단점

- **장점** :

① R-CNN의 많은 문제점 (복잡한 파이프라인, 여러개의 분류기를 트레이닝해야하는 문제점, 느린 추론 시간 등)을 개선함

- **단점** :

① 여전히 Selective Search 수행에 2초가 걸려서 1장의 이미지에 약 2.3초의 추론시간이 필요함

Thank you!