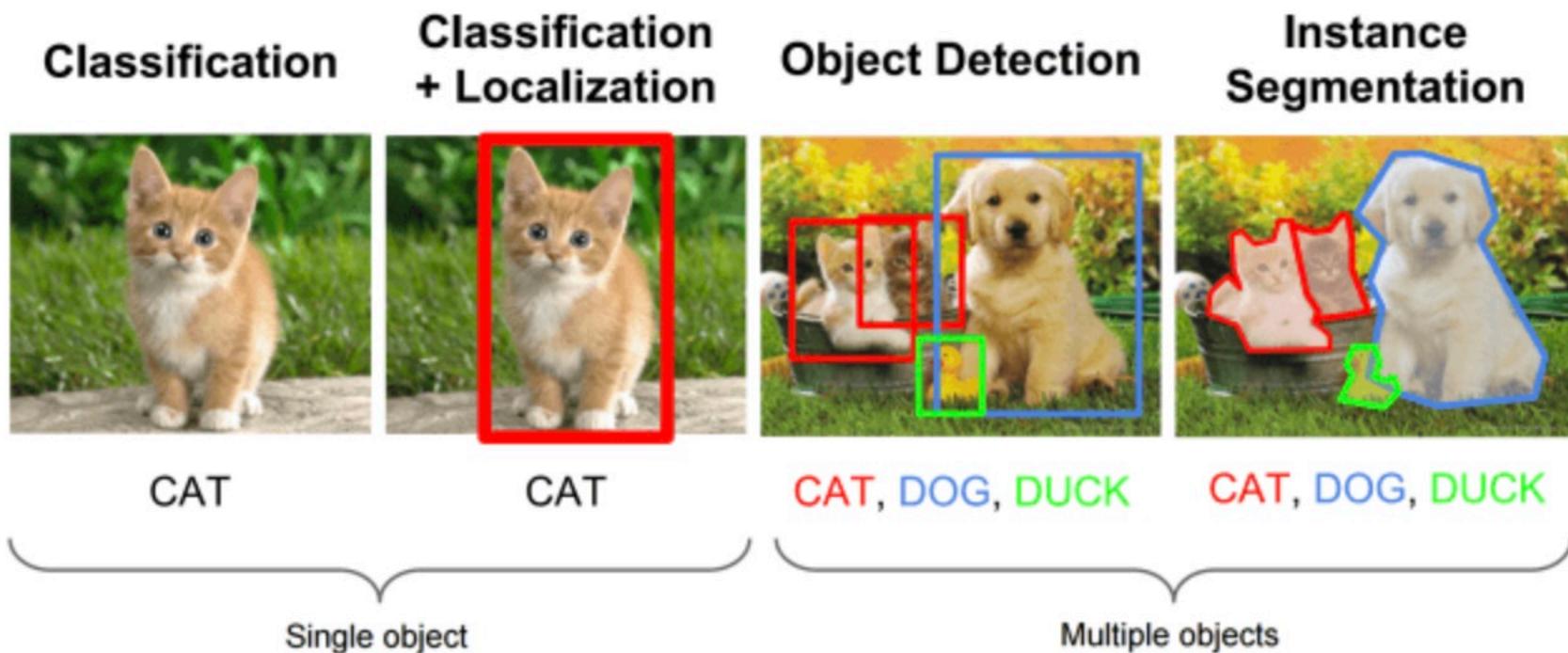


Object Detection

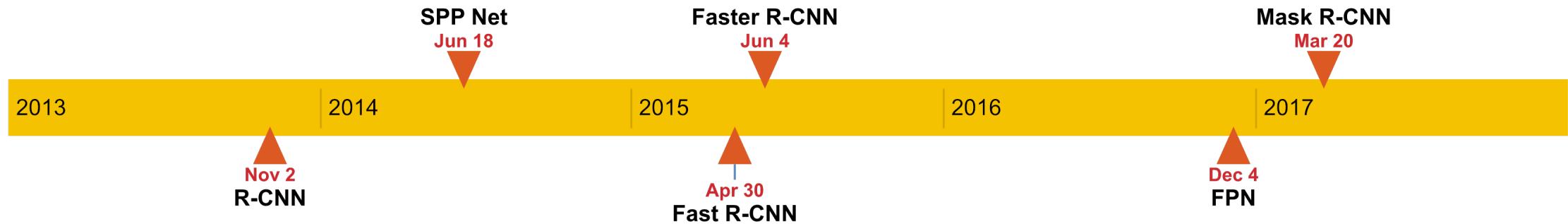
from R-CNN to Faster R-CNN

zhs2326

Object Detection



R-CNN (Region-based Convolutional Neural Networks)

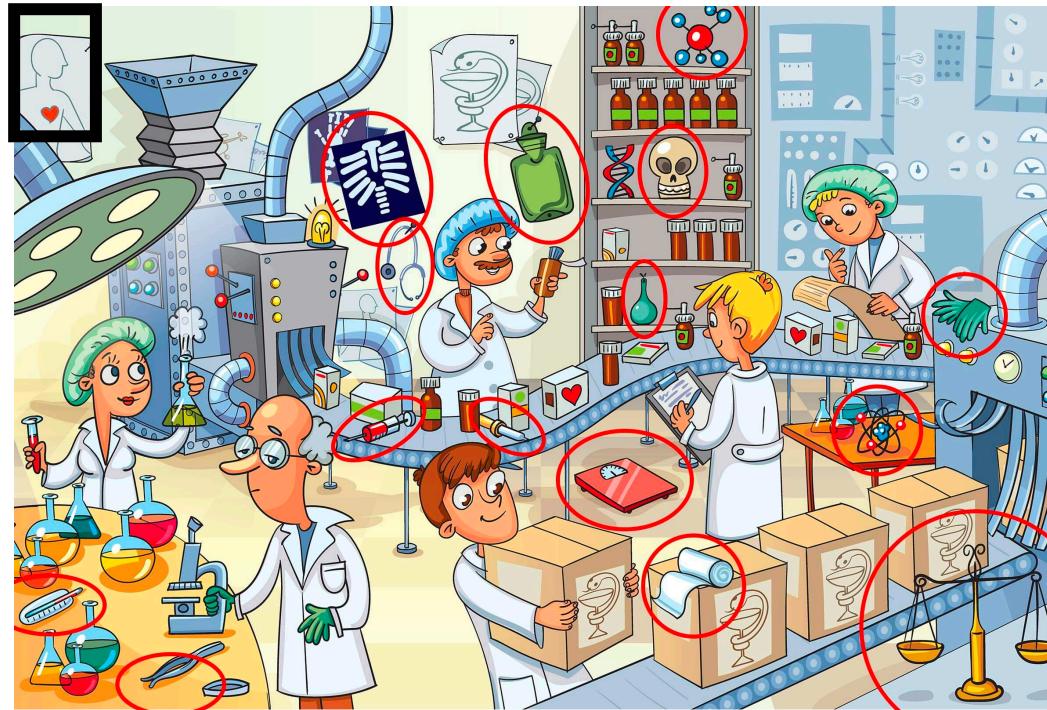


Ross Girshick et al. **Rich feature hierarchies for accurate object detection and semantic segmentation** (CVPR 2014)

<https://github.com/rbgirshick/rcnn>

R-CNN

Simplest Method —— Sliding Window

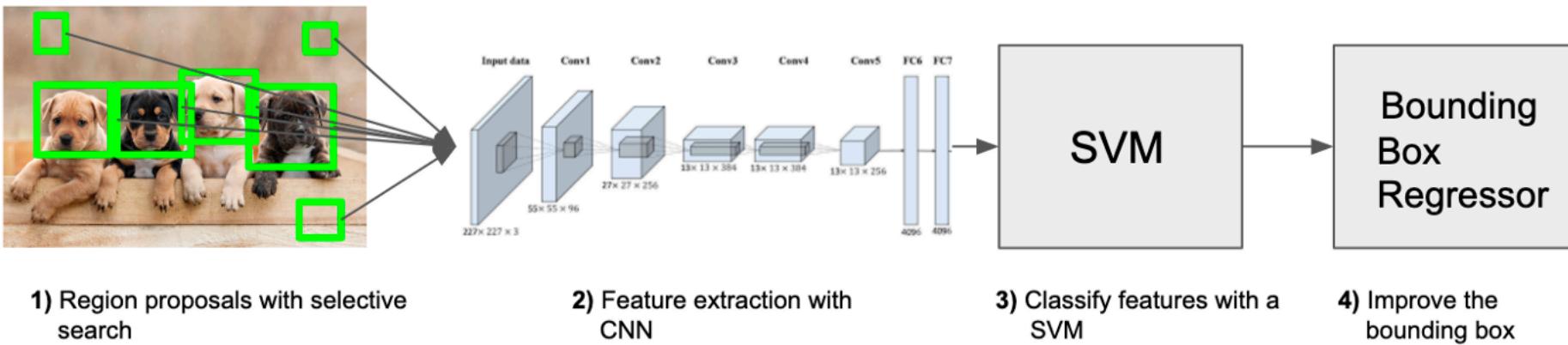


Find 15 objects in the picture



R-CNN

Intuition: Region Proposal + CNN



R-CNN

Details

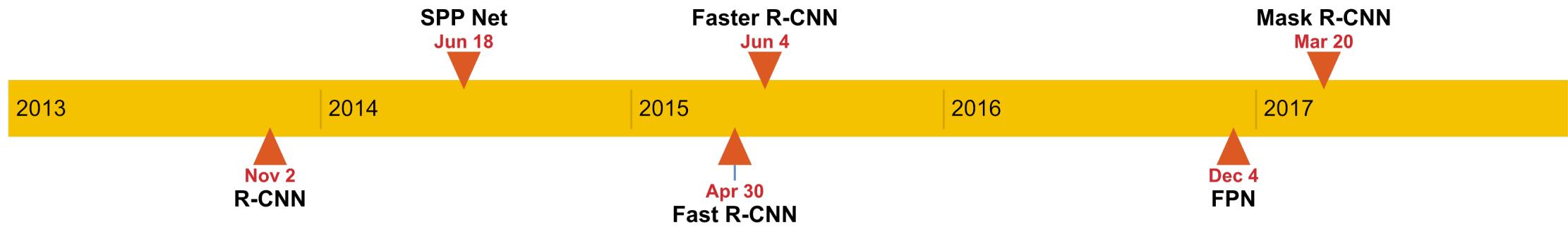
- 2000 region proposals by selective search
- Resize all region proposals to 227*227 (AlexNet)
- Why SVM instead of fully connected layers?(mAP decreases from 54% to 51%)

R-CNN (Slow R-CNN)

Drawbacks

- Slow(training time: 84 hours, test time per image: 47 (50) seconds)
- 2000 region proposals by selective search
- Training is a multi-stage pipeline(cnn, classification, regression)
- CNN(log loss), classification(hinge loss), regression(least squares)
- Classification and regression are post-hoc

Fast R-CNN



Ross Girshick **Fast R-CNN** (ICCV 2015)
<https://github.com/rbgirshick/fast-rcnn>

Fast R-CNN

Drawbacks of R-CNN:

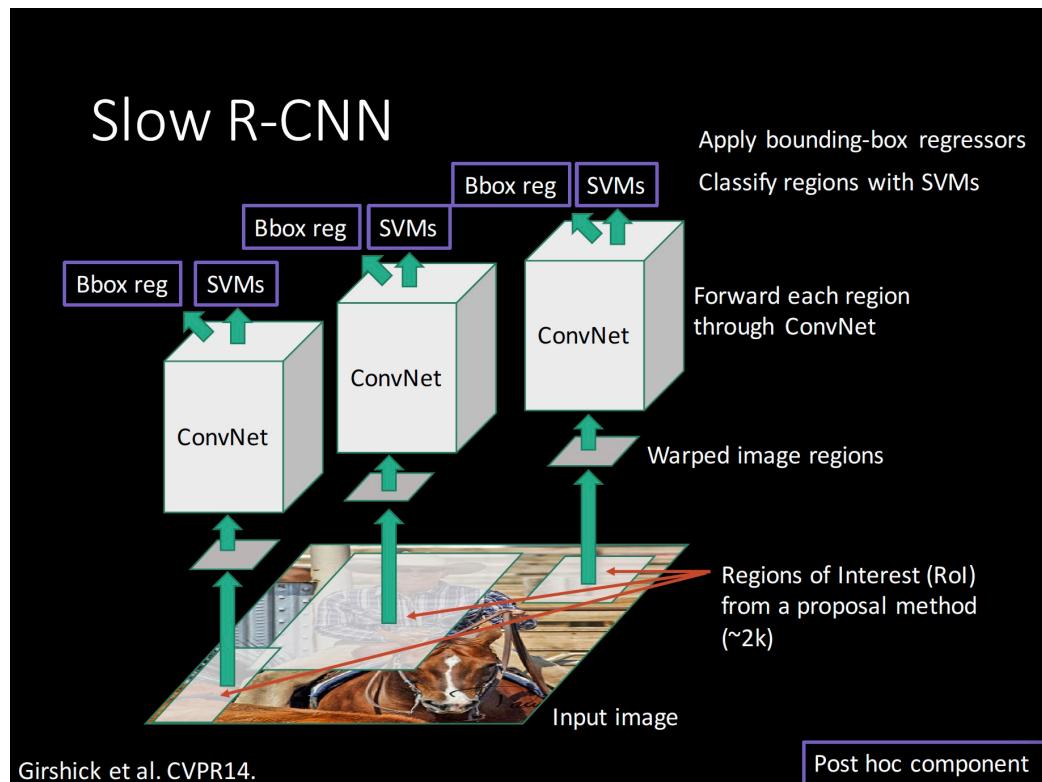
Independent forward pass of CNN on every region proposal

Complex pipeline/post-hoc training

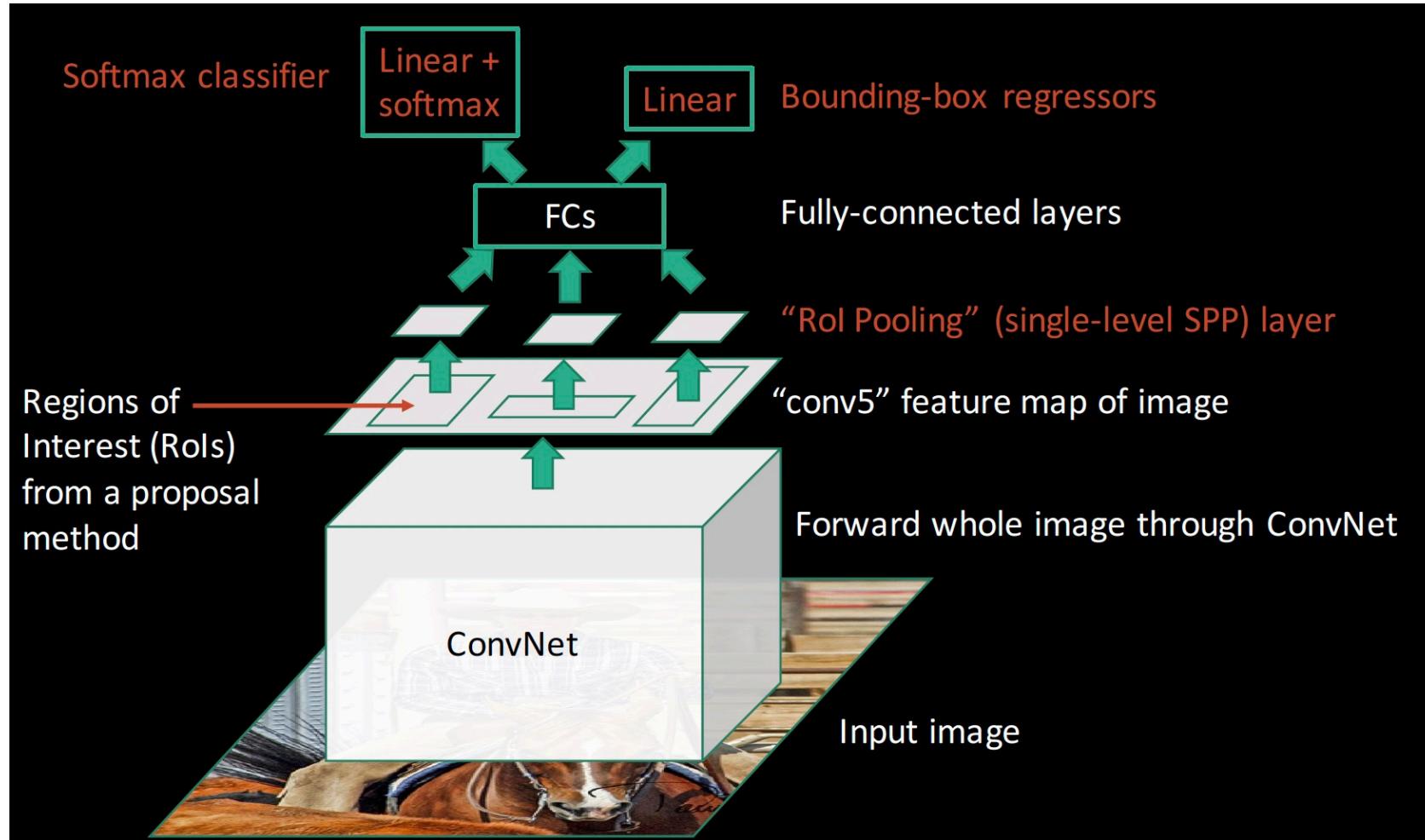
Solutions:

Share CNN between proposals

Train an end-to-end system



Fast R-CNN



Fast R-CNN

Details:

- ROI pooling layer (fully connected layers require fixed input size)
- Multitask loss

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

- Hierarchical mini-batch sampling (size=128, N=2, R=64)
- Truncated SVD for faster detection

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

- Why softmax instead of SVM?

Fast R-CNN

Why fast?

- just one single time CNN forward pass
- selective search can be done in parallel
- Truncated SVD

Training time: 84h -> 9.5h

Test time: 47s -> 0.32 (2) s

Test time: 0.32s -> 0.22s

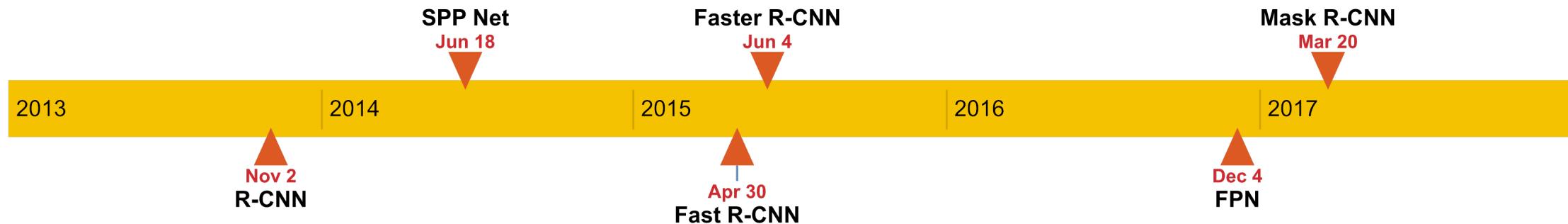
mAP: 66.9 -> 66.6

Fast R-CNN (Fast Slow R-CNN)

Drawbacks:

- Selective Search! (too slow, only on cpu)
- $0.32s \rightarrow 2s$

Faster R-CNN



Shaoqing Ren et al. **Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks** (NIPS 2015)

<https://github.com/rbgirshick/py-faster-rcnn>

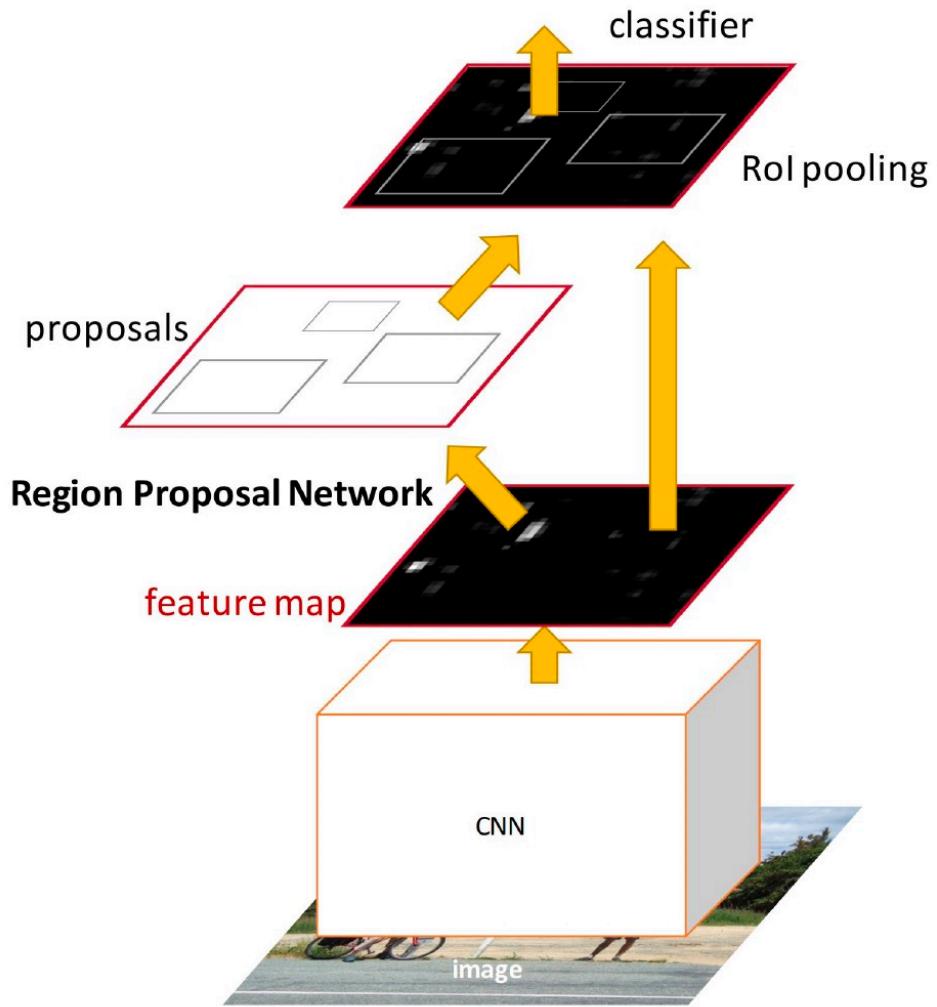
<https://github.com/facebookresearch/detectron2>

Faster R-CNN

Intuition:

- Improve the bottleneck (region proposal)
- Let CNN do the region proposal

Faster R-CNN



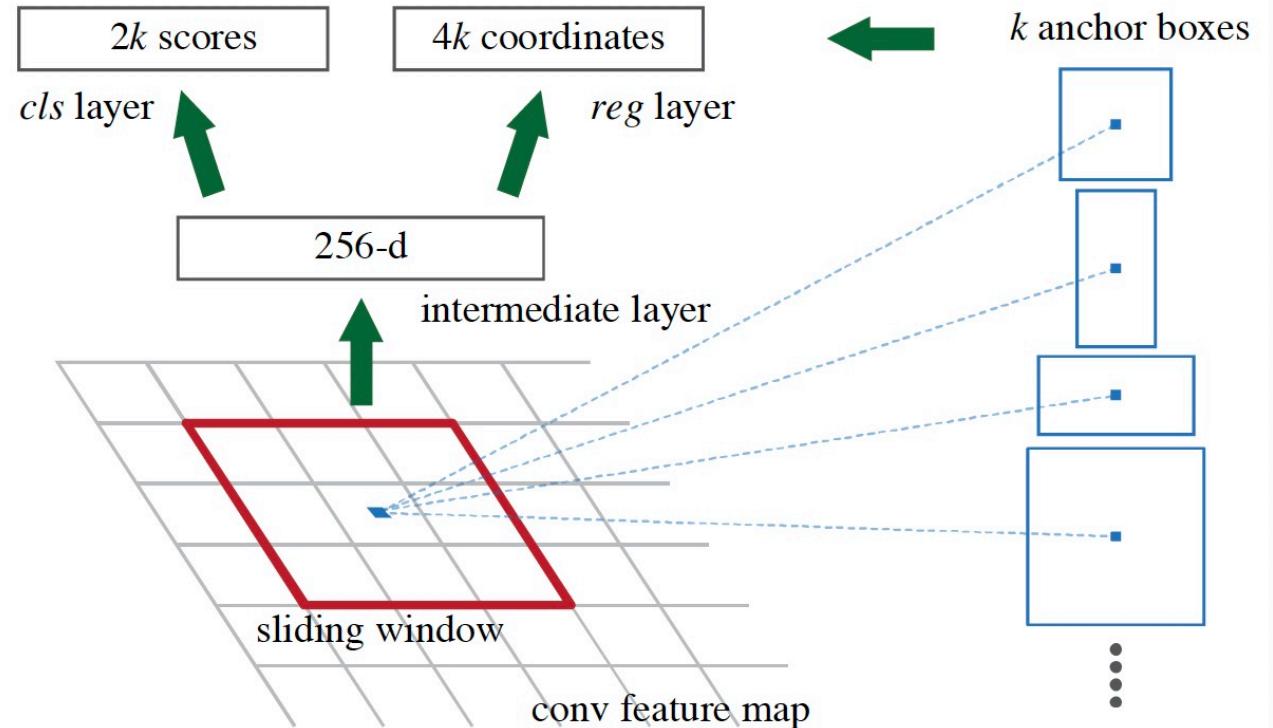
RPN + Fast R-CNN

Faster R-CNN

RPN (Region Proposal Network)

- 9 anchor boxes (3 scales, 3 aspect ratio)
- Classify objects or not
- Multi-task loss

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



Faster R-CNN

- Test time: 2s -> 0.2s

Faster Fast Slow R-CNN

R-CNN family (two-stage)

VS

YOLO family (one-stage)

Thank You