

# Fast RCNN

## In depth look

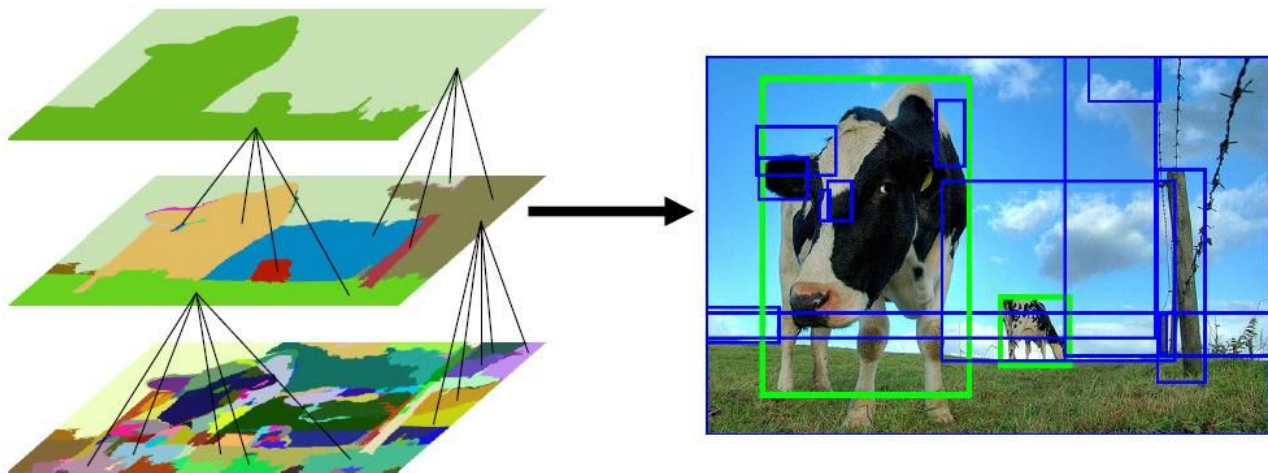
Błażej Osiński

# Object Detection

- Region proposal
- Region evaluation: class and bounding box

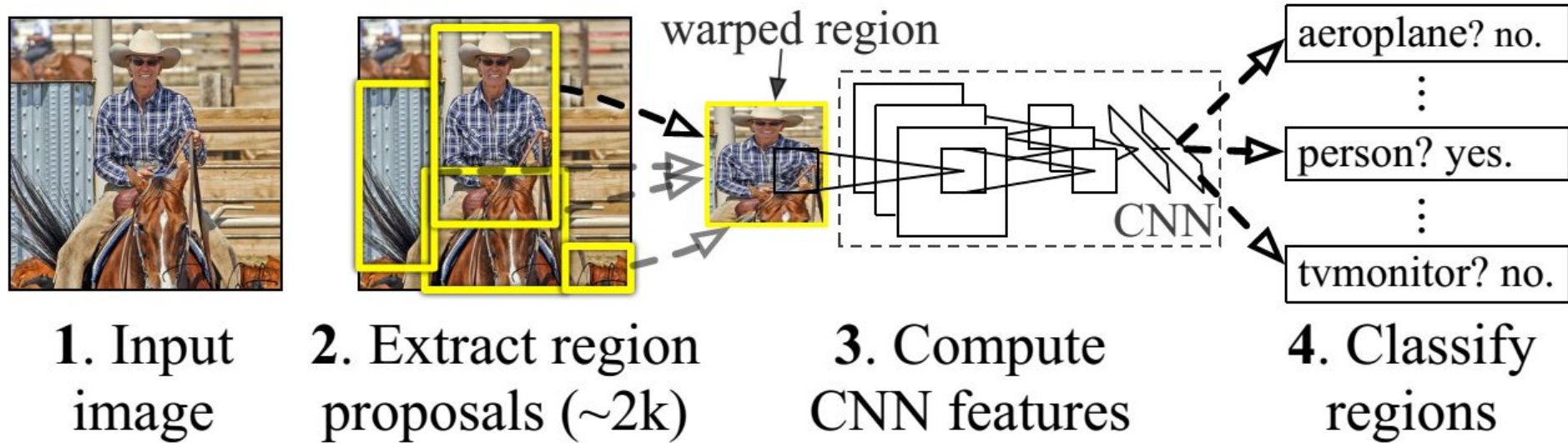
After: [A seismic shift in object detection](#) December 2013

- Sliding window  
→ Region proposal  
by Segmentation
  - Selective search
  - Edge boxes



# 2014 - Region-based Convolutional Neural Networks

## **R-CNN: *Regions with CNN features***



# R-CNN ~~drawbacks~~ challenges

- Complicated pipeline
- Storing a lot of stuff on the disc
- Slow training and testing

# 2015 - Fast R-CNN

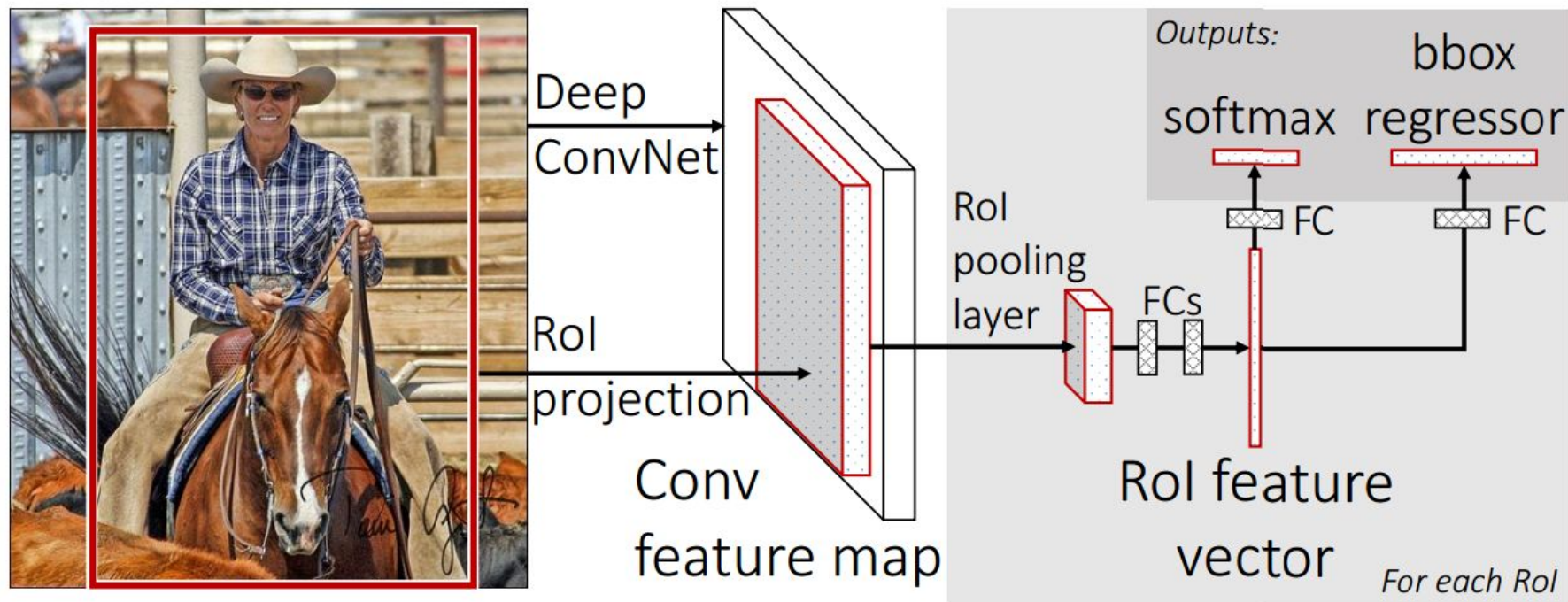


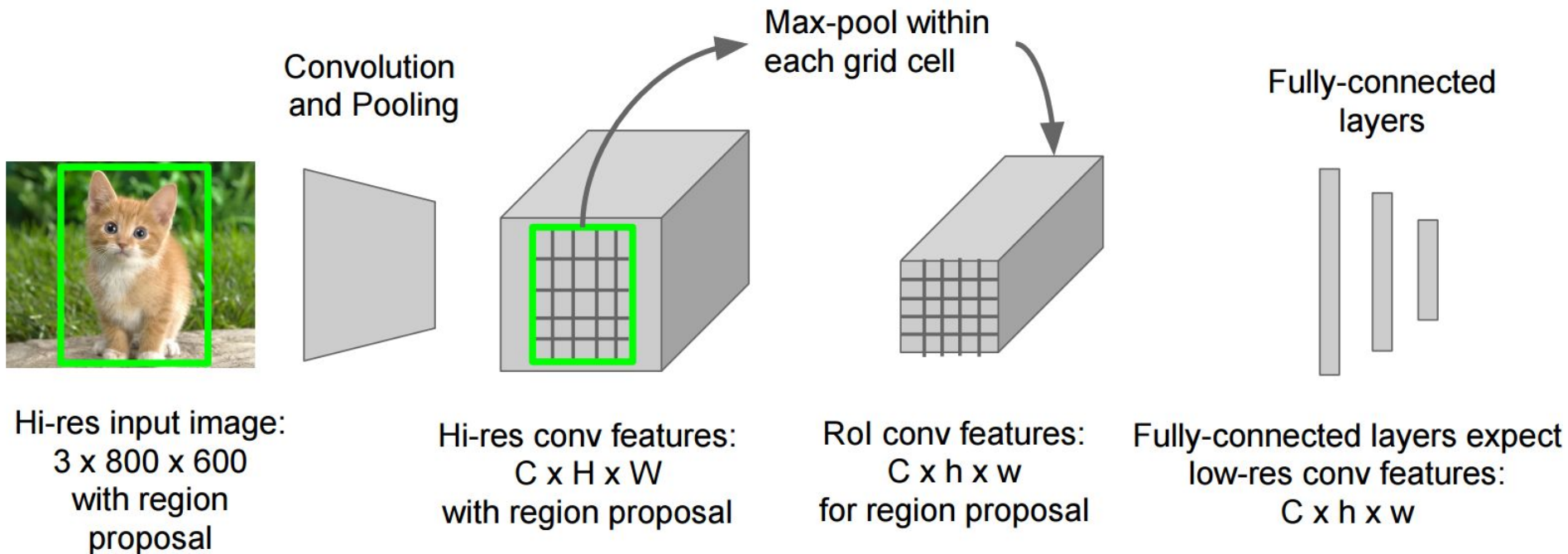
Figure 1. Fast R-CNN architecture. An input image and multi-

# Initializing from pre-trained networks

Pretrain network, e.g. VGG16

- Last max pooling → RoI Pooling
- Last FC & Softmax →
  - FC & Softmax ( $K + 1$  categories)
  - Category specific regressor
- Two inputs → image and region proposals

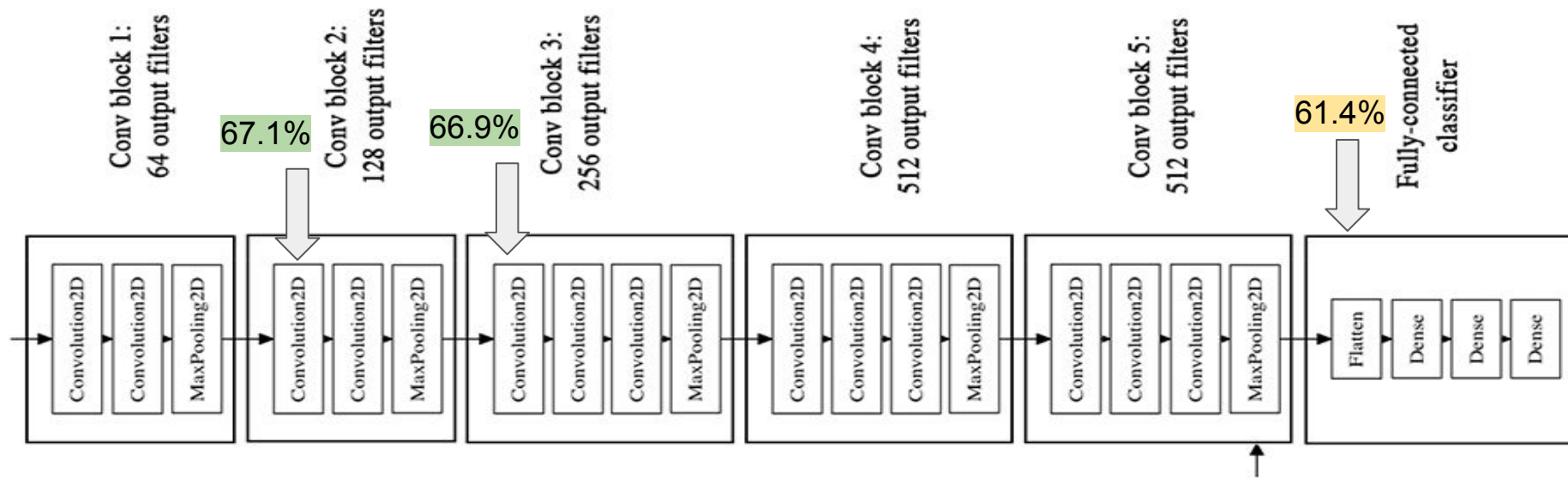
# Region of Interest (RoI) Pooling



Slide credit: CS231, Fei-Fei Li & Andrej Karpathy & Justin Johnson

# Training Region of Interest (RoI) Pooling

- Batch size
  - 2 images
  - 64 region proposals per image (25% intersected with objects)
- Training depth





# mAP - mean average precision

- For each class compute AP
- $AP = \text{AUC of precision / recall for all detection from all images}$
- True positive =  $\text{IoU} > \text{threshold (usually 0.5)}$

# How fast is Fast R-CNN?

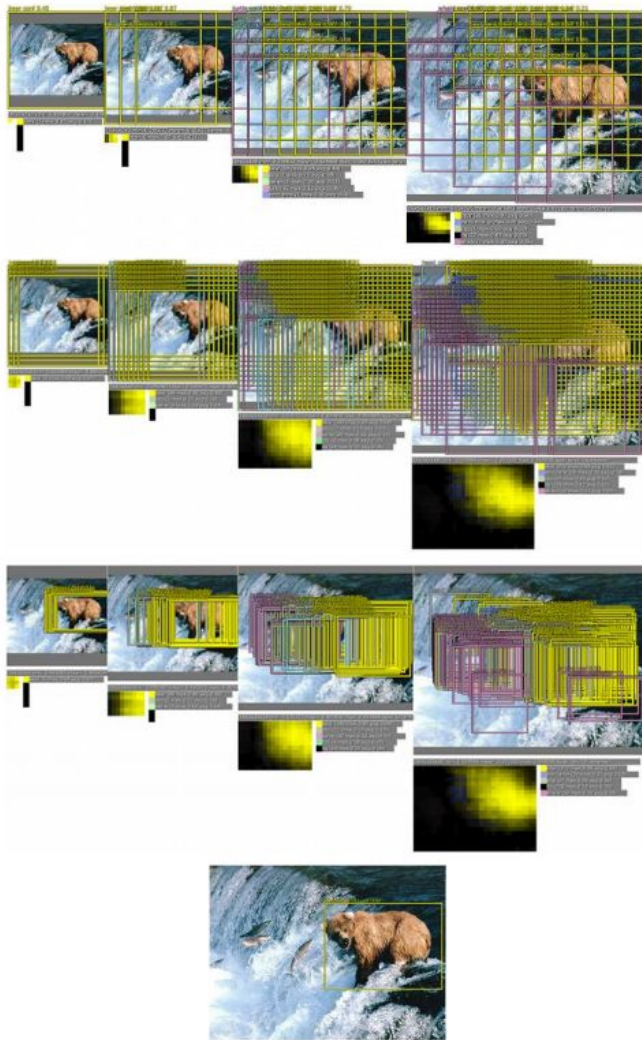
	R-CNN	Fast R-CNN
Training time	84 h	9.5 h
Test time per image	47 s	0.32 s
Test time per image With Selective Search	50 s	2 s



# OverFeat

Integrated Recognition, Localization  
and Detection using Convolutional Networks

[arxiv](https://arxiv.org/abs/1411.9001)



# 2015 - Faster R-CNN

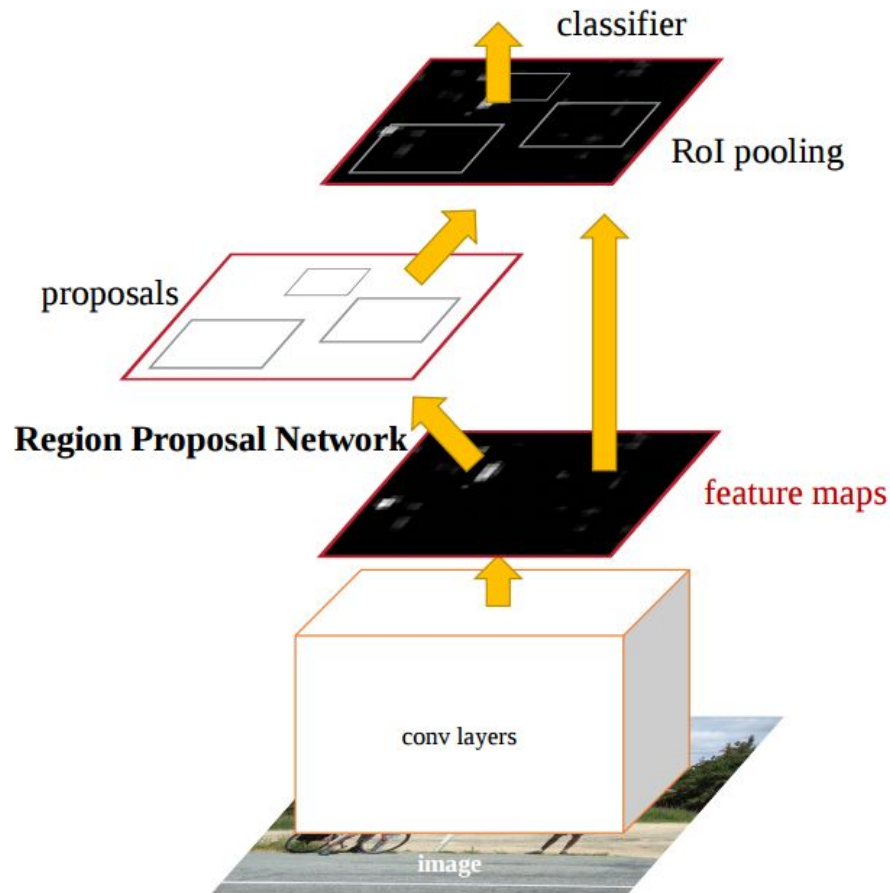


Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.