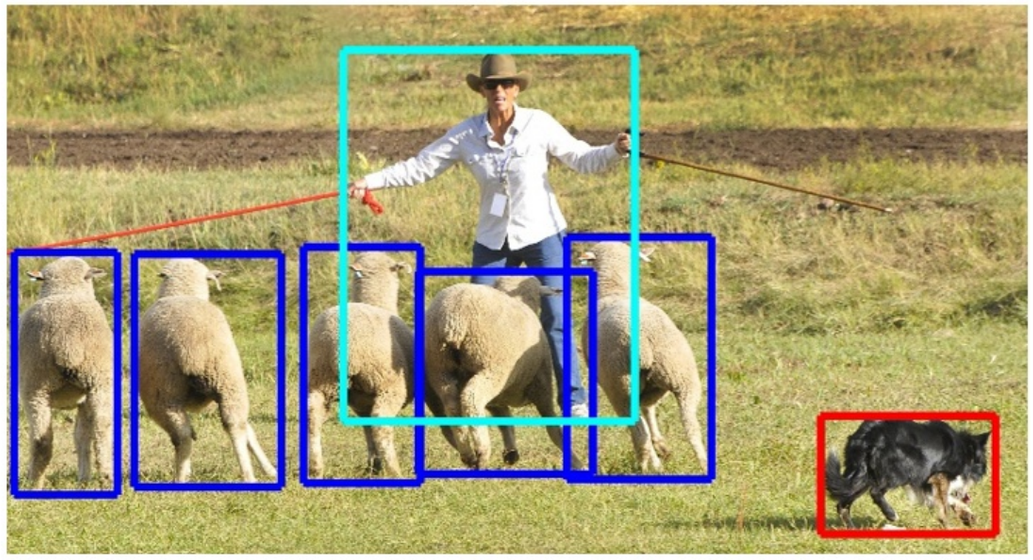
OBJECT DETECTION AND SEGMENTATION

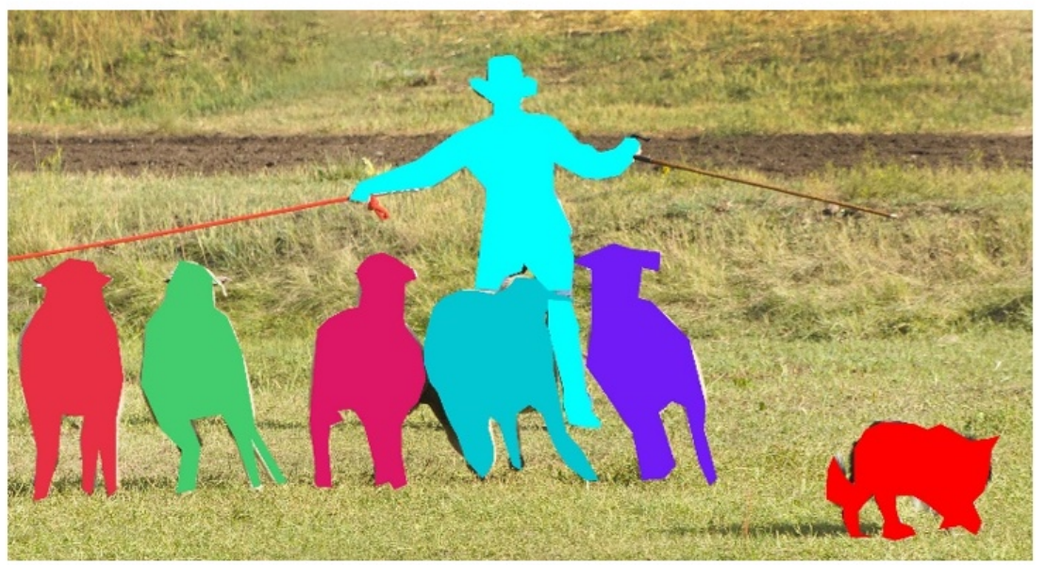
DETECTION/SEGMENTATION TASKS

(a) Image classification



(b) Object localization

(c) Semantic segmentation



(d) This work

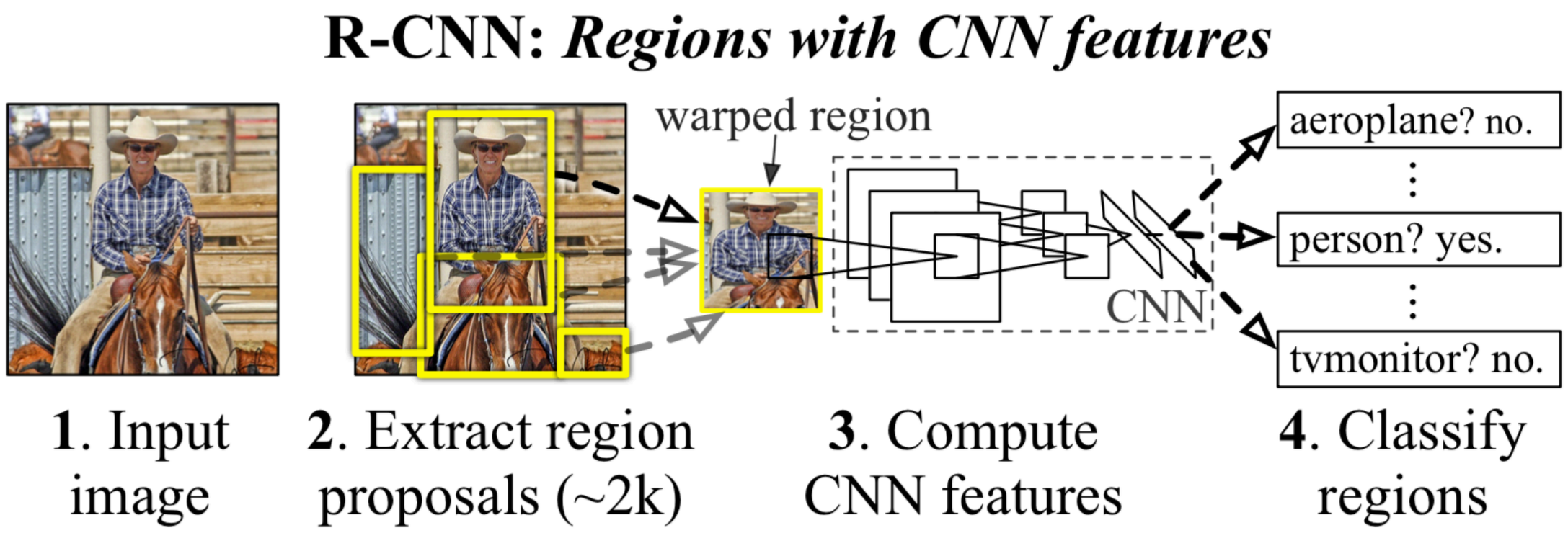
BEYOND CLASSIFICATION

- 2012-2014: classification works, but …
- CNN-s lose spatial details (max-pool), they are supposed the be somewhat “spatially invariant”
- Can it predict precise locations too?
- Can it provide accurate pixel masks?

R-CNN

REGIONS WITH CONVOLUTIONAL NEURAL NETWORKS

- Girshick, Ross, et al. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2014. (~8000 citations)
- Regions with a bottom up algorithm: *Uijlings, Jasper RR, et al. "Selective search for object recognition*.2013
- Cut out 2000 proposed regions, warp them to 224x224, extract CNN features, classify with a linear SVM

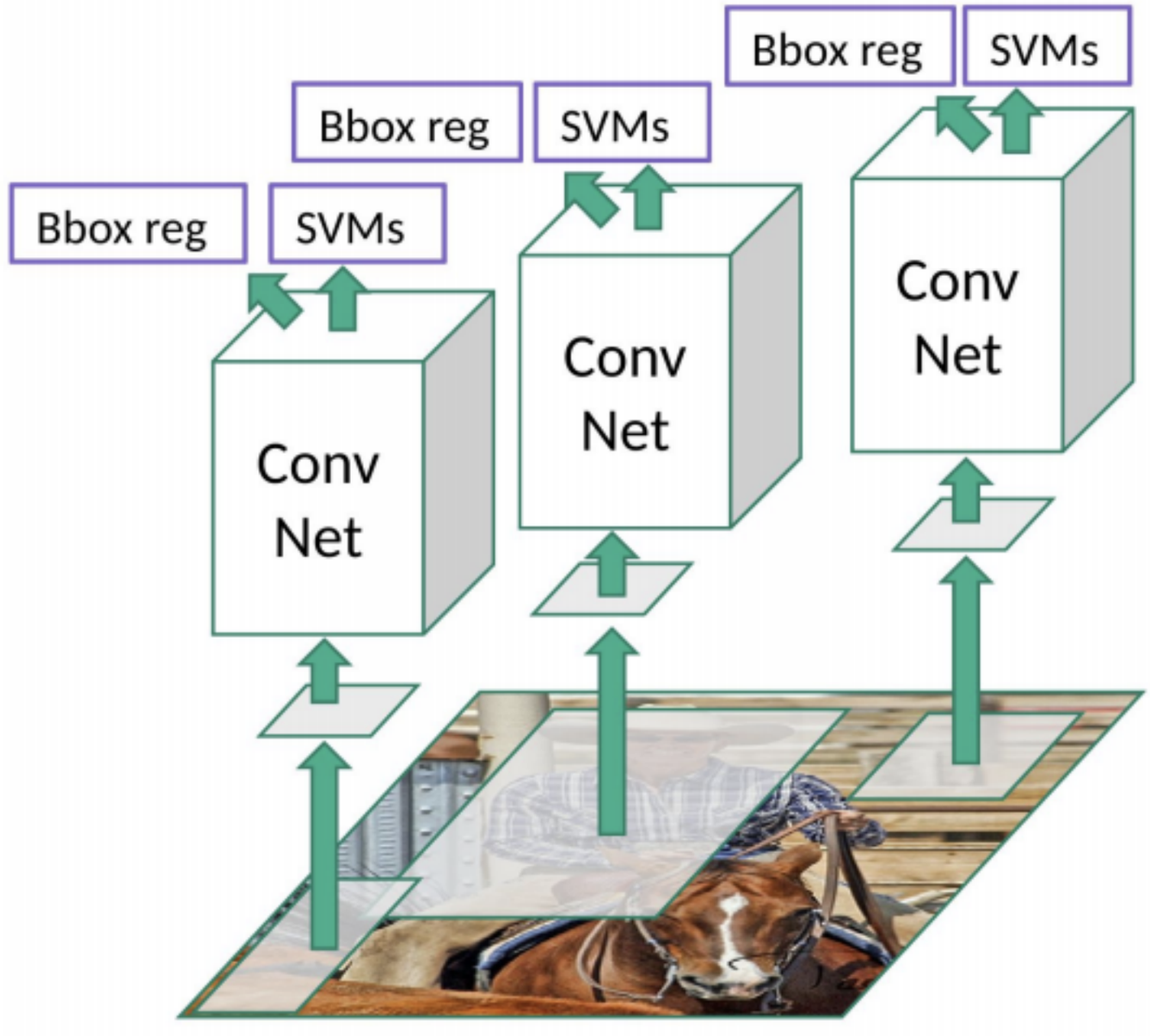


REGIONS WITH CONVOLUTIONAL NEURAL NETWORKS

- Girshick, Ross, et al. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2014. (~8000 citations)

- Regions with a bottom up algorithm: *Uijlings, Jasper RR, et al. "Selective search for object recognition*.2013

- Cut out 2000 proposed regions, warp them to 224x224, extract CNN features, classify with a linear SVM



REGIONS WITH CONVOLUTIONAL NEURAL NETWORKS

- *We discriminatively pre-trained the CNN on a large auxiliary dataset (ILSVRC 2012) with image-level annotations (i.e., no bounding box labels).*
- *To adapt our CNN to the new task (detection) and the new domain (warped VOC windows), we continue stochastic gradient descent (SGD) training of the CNN parameters using only warped region proposals …. replacing the CNN's ImageNet-specific 1000-way classification layer with a randomly initialized 21-way classification layer*
- *We treat all region proposals with ≥ 0.5 IoU over- lap with a ground-truth box as positives for that box's class and the rest as negatives.*
- *We start SGD at a learning rate of 0.001 (1/10th of the initial pre-training rate), which al- lows fine-tuning to make progress while not clobbering the initialization.*
- *we uniformly sample 32 positive windows (over all classes) and 96 background windows to construct a mini-batch of size 128. We bias the sampling towards positive windows because they are extremely rare compared to background.*

REGIONS WITH CONVOLUTIONAL NEURAL NETWORKS

- *Since the training data is too large to fit in memory, we adopt the standard hard negative mining method*
- *we train a linear regression model to predict a new detection window given the pool5 features for a selective search region proposal.*
- *we apply a greedy non-maximum suppression (for each class independently) that rejects a region if it has an intersection-over-union (IoU) overlap with a higher scoring selected region larger than a learned threshold.*

REGIONS WITH CONVOLUTIONAL NEURAL NETWORKS

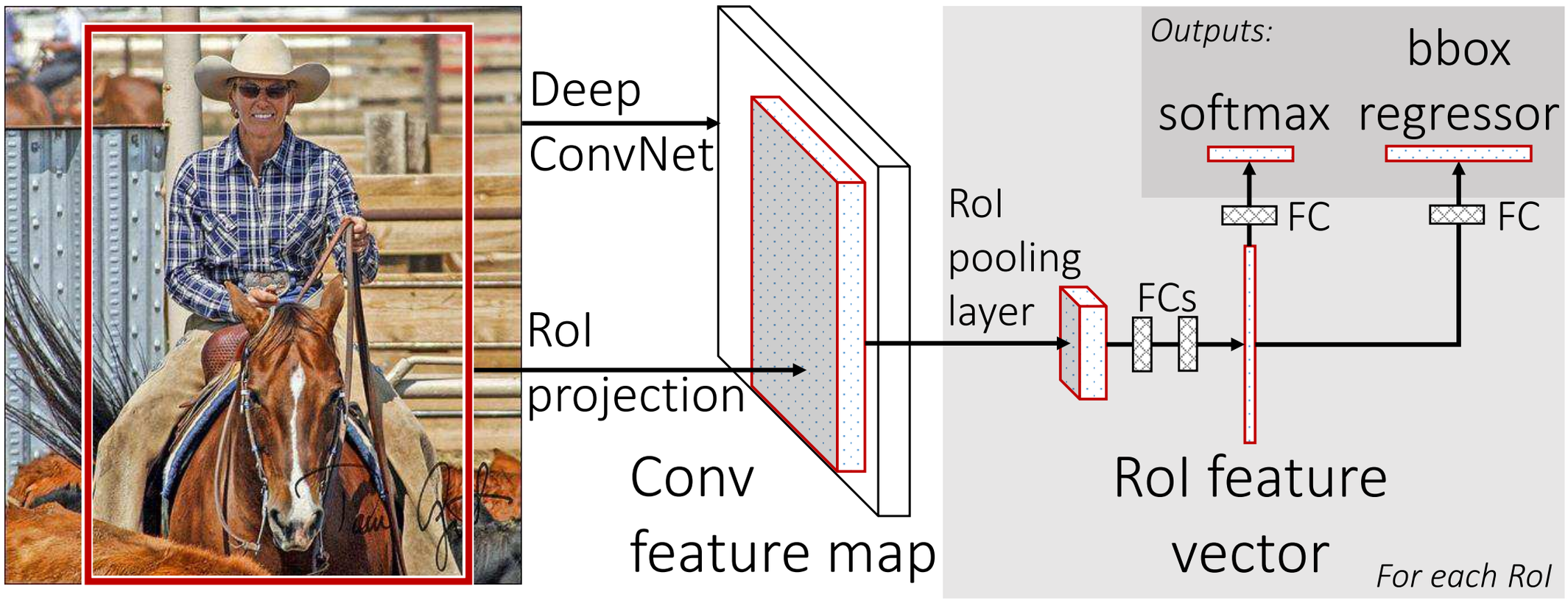
- *Can we gain insight into the representation learned by the CNN? Perhaps the densely connected layers, with more than 54 million parameters, are the key? They are not. We "lobotomized" the CNN and found that a surprisingly large proportion, 94%, of its parameters can be removed with only a moderate drop in detection accuracy.*

- *We conclude by noting that it is significant that we achieved these results by using a combination of classical tools from computer vision and deep learning (bottom- up region proposals and convolutional neural networks). Rather than opposing lines of scientific inquiry, the two are natural and inevitable partners.*

FAST R-CNN

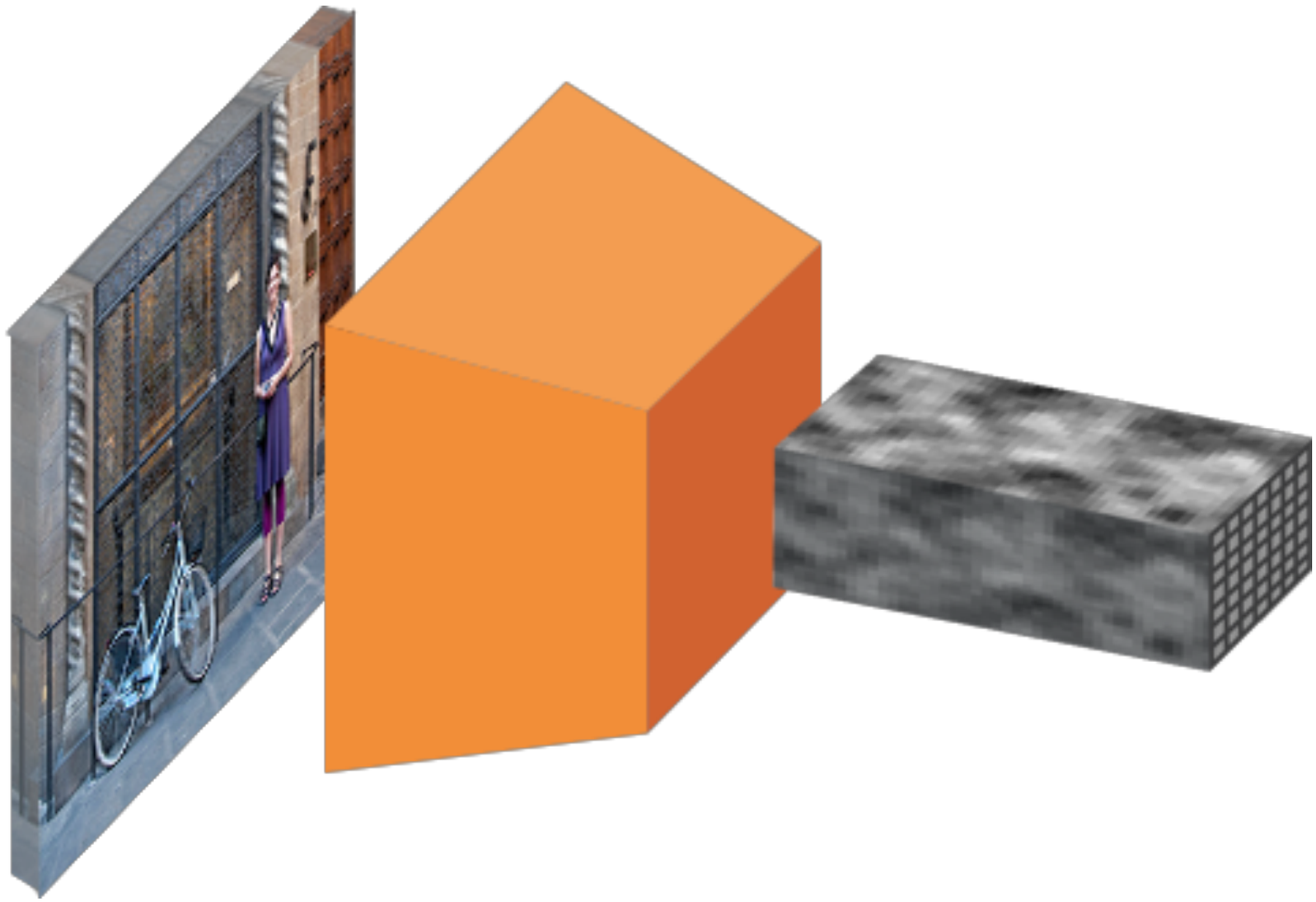
FAST R-CNN

- Girshick, Ross. "Fast r-cnn." (2015). (~5000 citations)
- R-CNN is slow: 2000 warped regions are evaluated on the GPU! For each class (21) a linear SVM with 4096 features is evaluated for these 2000 regions. ~1 minute then.
- 213 × faster at test-time, AND more accurate!



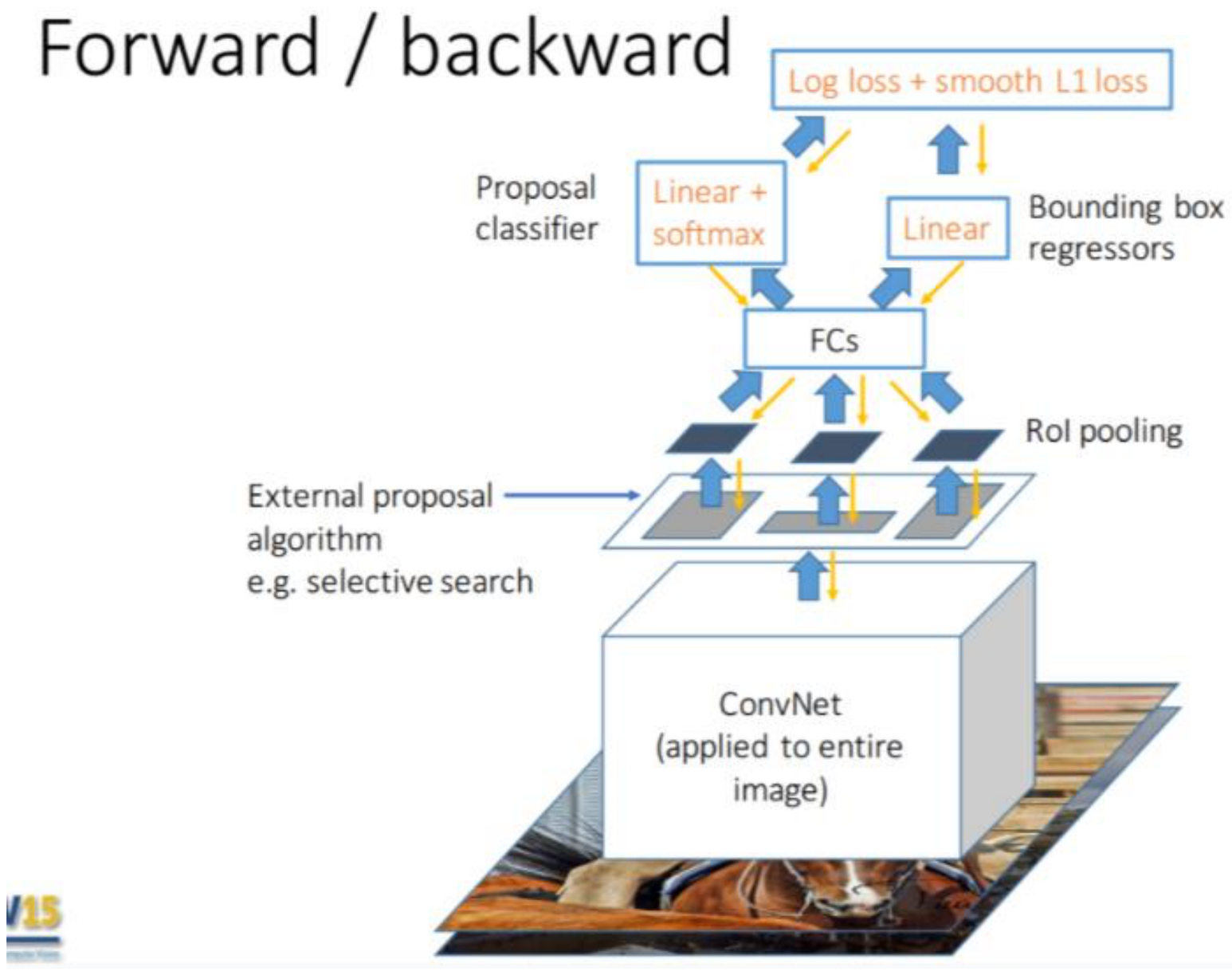
FAST R-CNN

- Girshick, Ross. "Fast r-cnn." (2015). (~5000 citations)
- R-CNN is slow: 2000 warped regions are evaluated on the GPU! For each class (21) a linear SVM with 4096 features is evaluated for these 2000 regions. ~1 minute then.
- 213 × faster at test-time, AND more accurate!



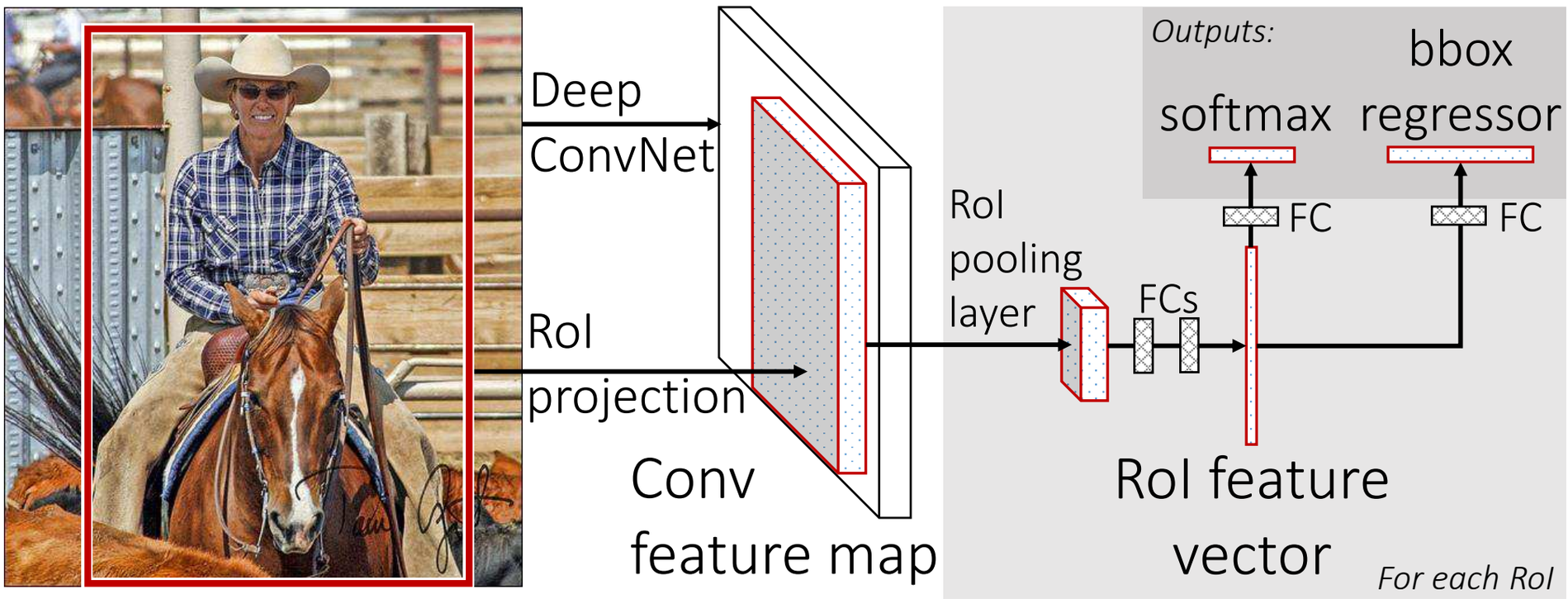
FAST R-CNN

- Girshick, Ross. "Fast r-cnn." (2015). (~5000 citations)
- R-CNN is slow: 2000 warped regions are evaluated on the GPU! For each class (21) a linear SVM with 4096 features is evaluated for these 2000 regions. ~1 minute then.
- 213 × faster at test-time, AND more accurate!



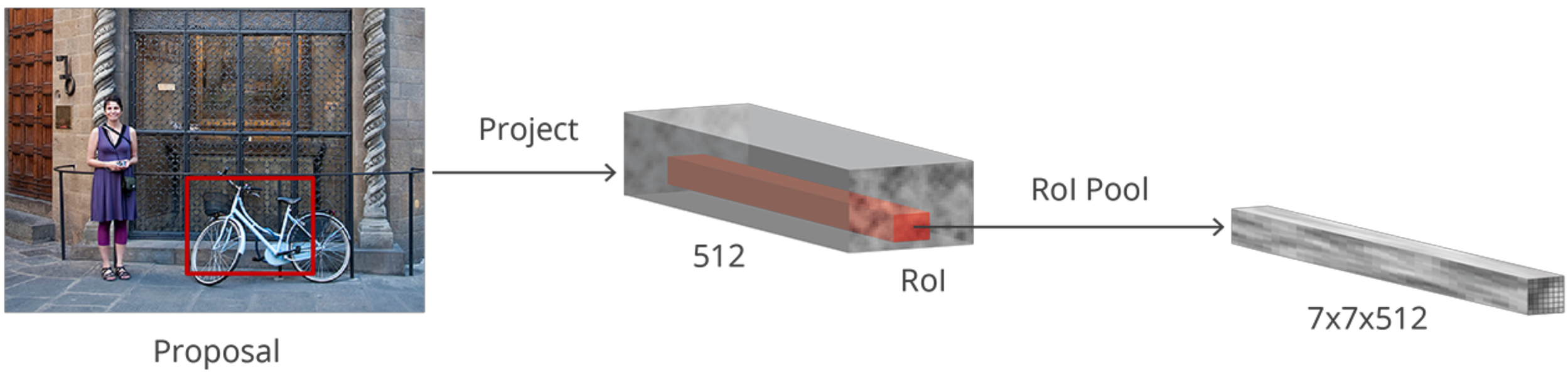
FAST R-CNN

- *The network first processes the whole image with several convolutional (conv) and max pooling layers to produce a conv feature map.*
- *Then, for each object proposal a region of interest (RoI) pooling layer extracts a fixed-length feature vector from the feature map.*
- *Each feature vector is fed into a sequence of fully connected (fc) layers that finally branch into two sibling output layers: one that produces softmax probability estimates over K object classes plus a catch-all "background" class and another layer that outputs four real-valued numbers for each of the K object classes. Each set of 4 values encodes refined bounding-box positions for one of the K classes.*



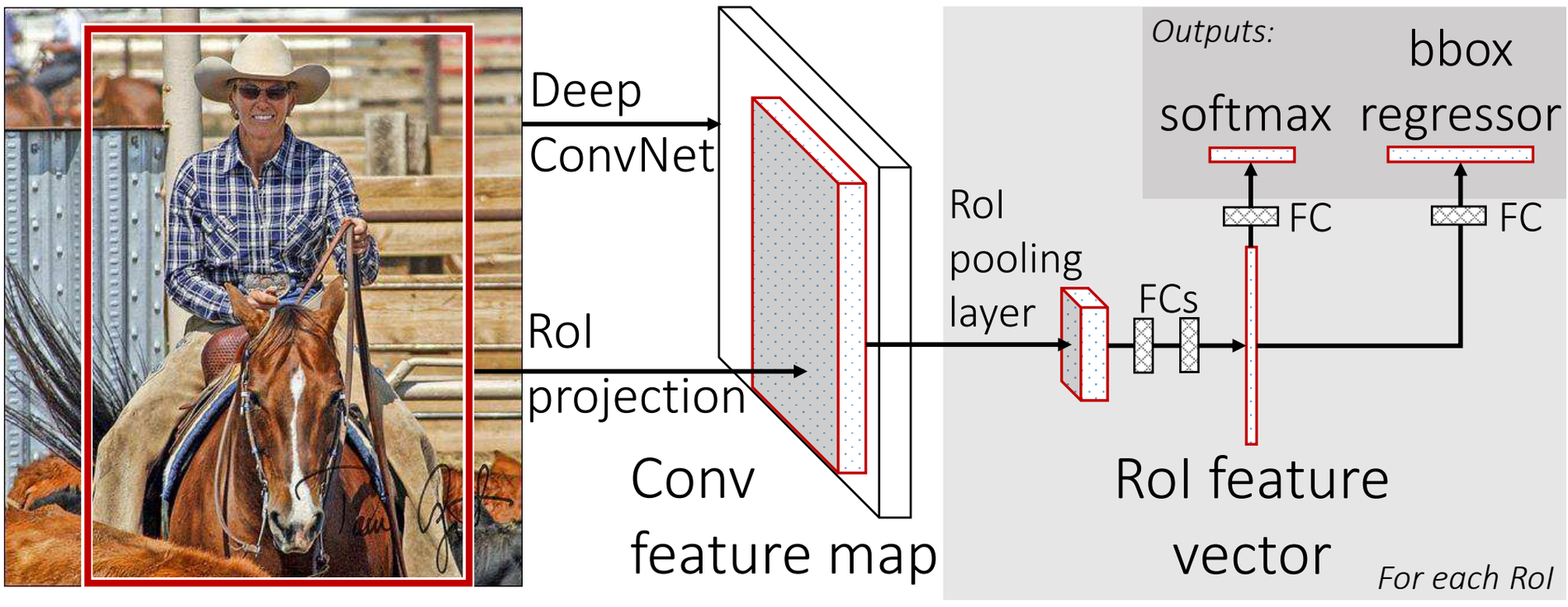
FAST R-CNN

- *The network first processes the whole image with several convolutional (conv) and max pooling layers to produce a conv feature map.*
- *Then, for each object proposal a region of interest (RoI) pooling layer extracts a fixed-length feature vector from the feature map.*
- *Each feature vector is fed into a sequence of fully connected (fc) layers that finally branch into two sibling output layers: one that produces softmax probability estimates over K object classes plus a catch-all "background" class and another layer that outputs four real-valued numbers for each of the K object classes. Each set of 4 values encodes refined bounding-box positions for one of the K classes.*



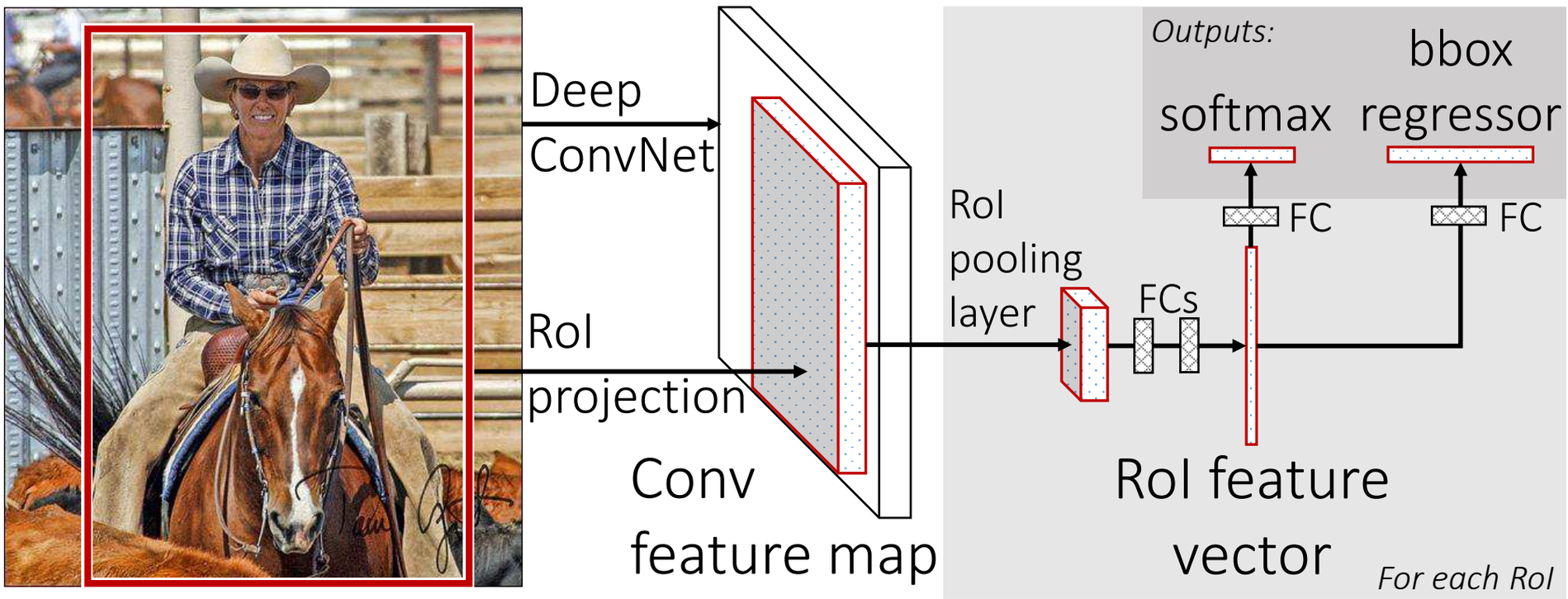
FAST R-CNN

- *RoI max pooling works by dividing the h × w RoI win- dow into an H × W grid of sub-windows of approximate size h/H × w/W and then max-pooling the values in each sub-window into the corresponding output grid cell.*



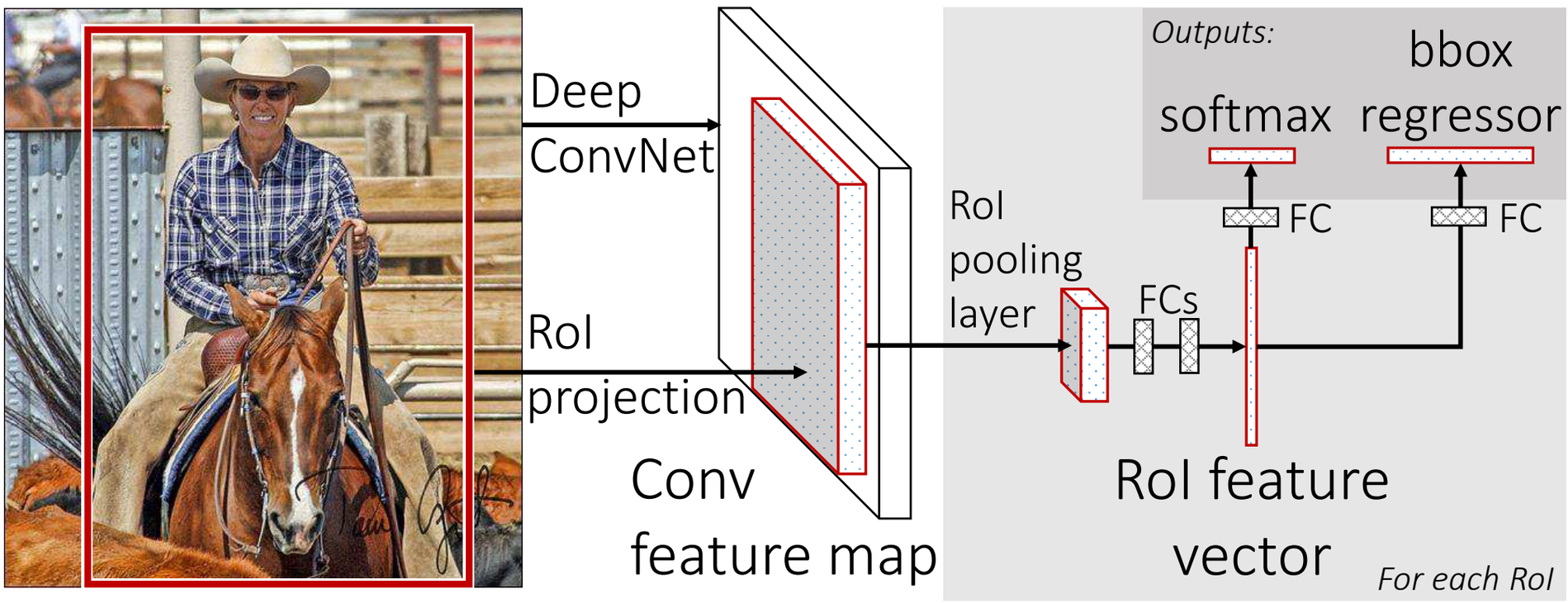
FAST R-CNN

- *The last max pooling layer is replaced by a RoI pooling layer that is configured by setting H and W to be compatible with the net's first fully connected layer (e.g., H =W =7 for VGG16).*
- *The network's last fully connected layer and softmax (which were trained for 1000-way ImageNet classification) are replaced with the two sibling layers described earlier (a fully connected layer and softmax over K + 1 categories and category-specific bounding-box regressors).*



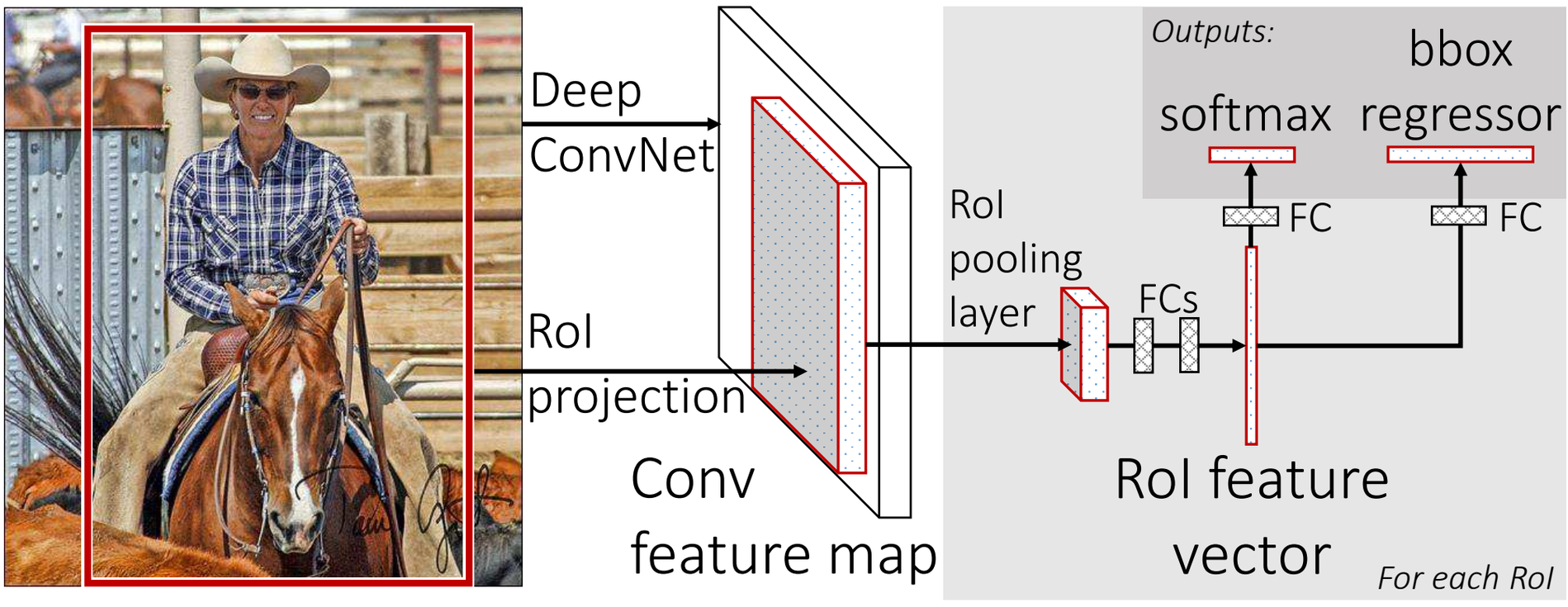
FAST R-CNN

- *Backprop through ROI pooling:*
- *A single input may be assigned to several different outputs.*
- *The RoI pooling layer's backwards function computes partial derivative of the loss function with respect to each input variable by following the argmax switches*
- *For each mini-batch RoI r and for each pooling output unit, the partial derivative is accumulated if i is the argmax selected for the output by max pooling.*



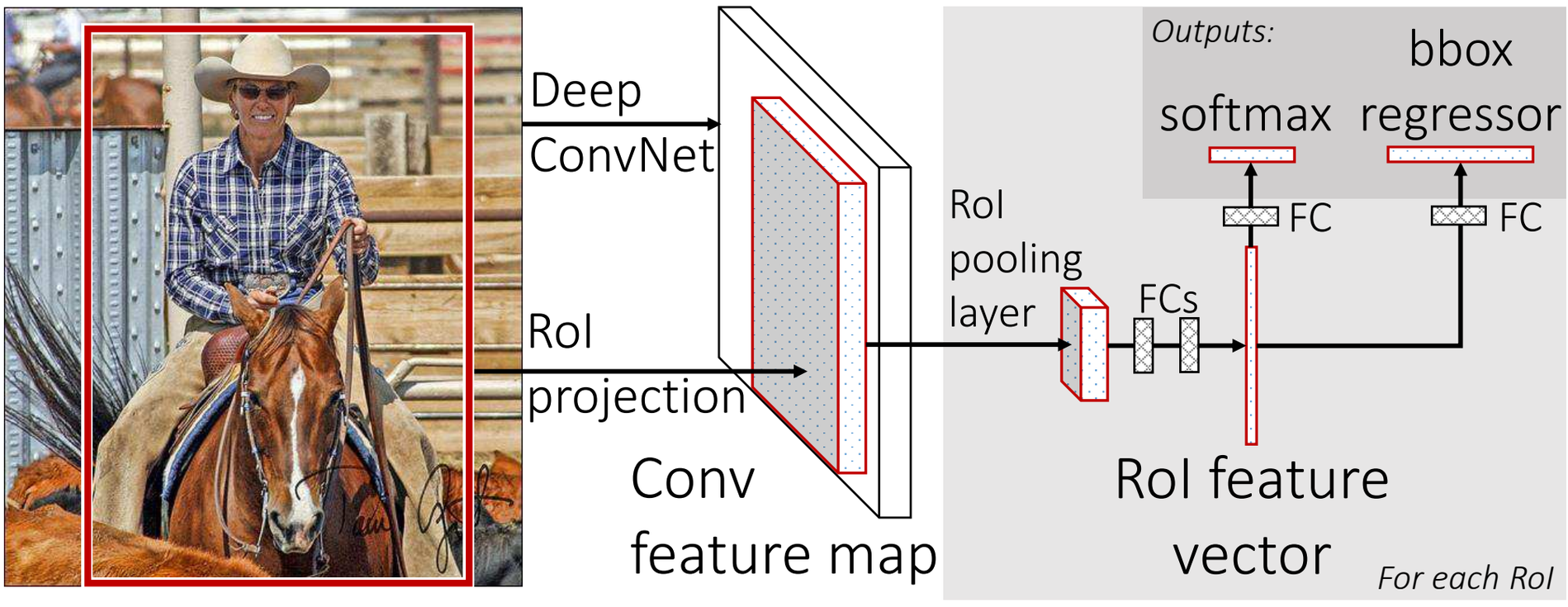
FAST R-CNN

- *Training all network weights with back-propagation is an important capability of Fast R-CNN*
- *Stochastic gradient descent (SGD) mini-batches are sampled hierarchically, first by sampling N images and then by sampling R/N RoIs from each image.*
- *RoIs from the same image share computation and memory in the forward and backward passes*
- *One concern over this strategy is it may cause slow train- ing convergence because RoIs from the same image are cor- related. This concern does not appear to be a practical issue*



FAST R-CNN

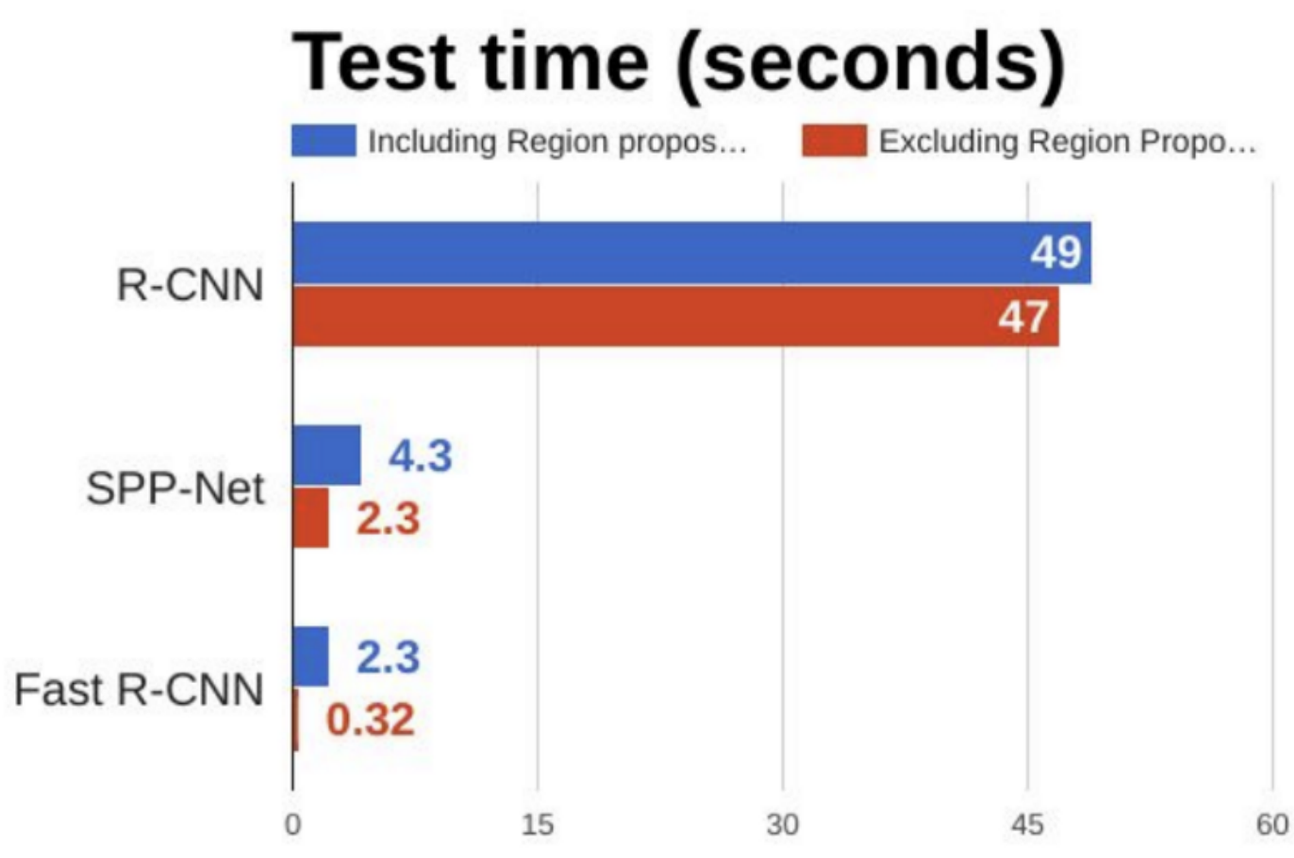
- *We take 25% of the RoIs from object proposals that have intersection over union (IoU) overlap with a ground- truth bounding box of at least 0.5.*
- *The remaining RoIs are sampled from object proposals that have a maximum IoU with ground truth in the interval [0.1, 0.5). These are the background examples and are labeled with u = 0. The lower threshold of 0.1 appears to act as a heuristic for hard example mining.*



FASTER R-CNN

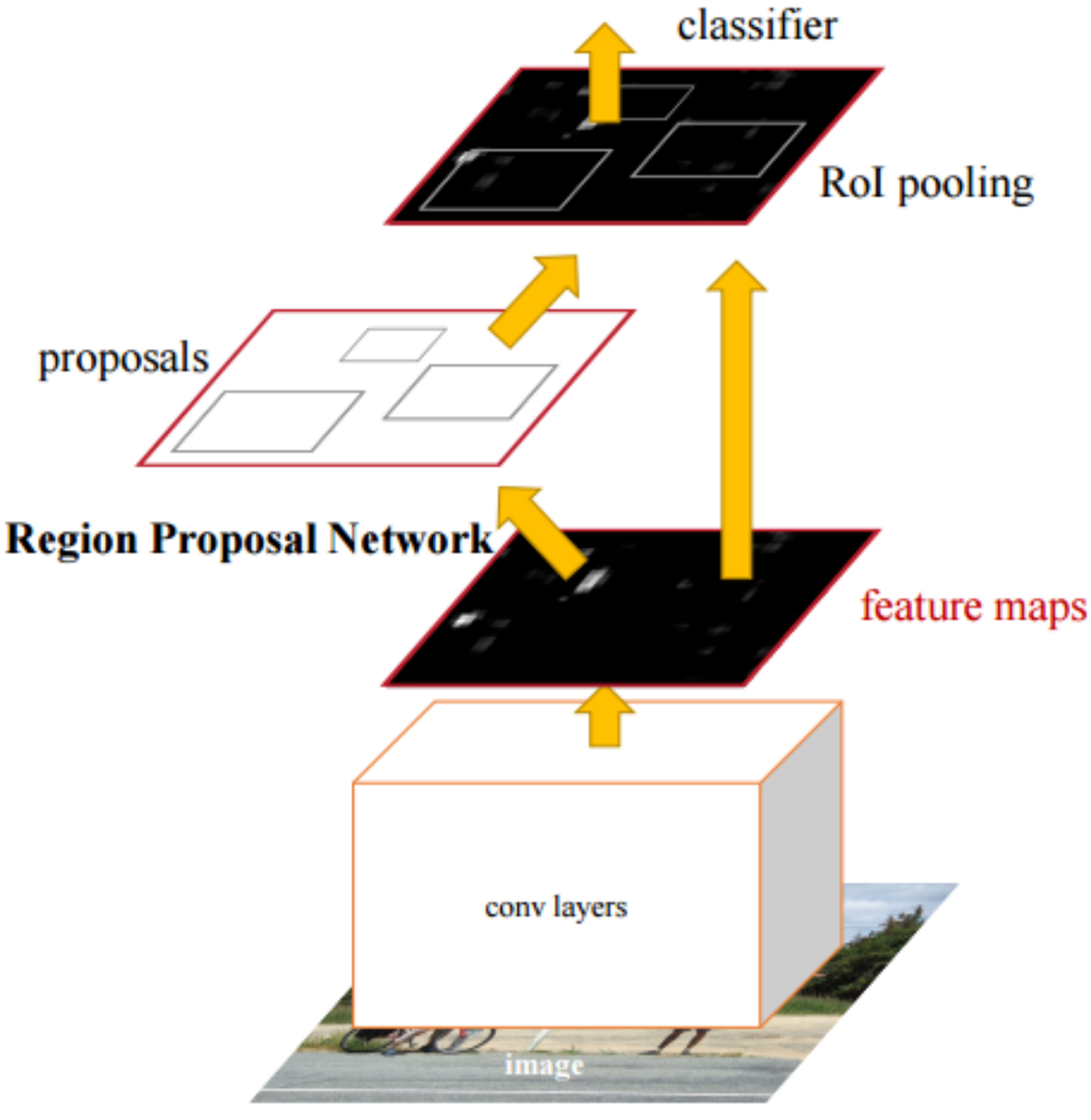
FASTER R-CNN

- Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." 2015. (~9000 citations)
- Fast R-CNN: region proposal by a classical algorithm is the slowest step.
- Faster: replaces classical algorithms for region proposals with the CNN.



FASTER R-CNN

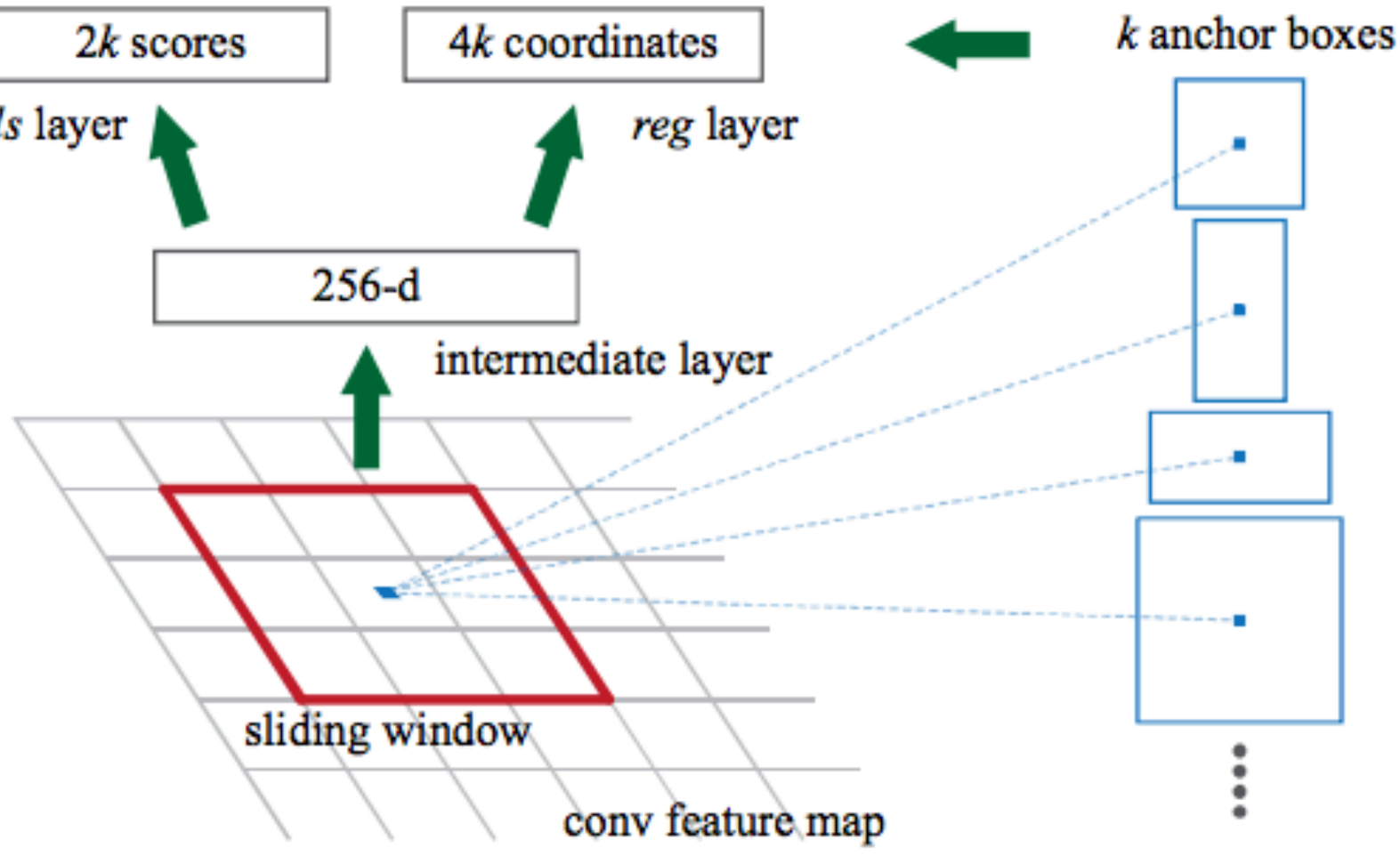
- Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." 2015. (~9000 citations)
- Fast R-CNN: region proposal by a classical algorithm is the slowest step.
- Faster: replaces classical algorithms for region proposals with the CNN.



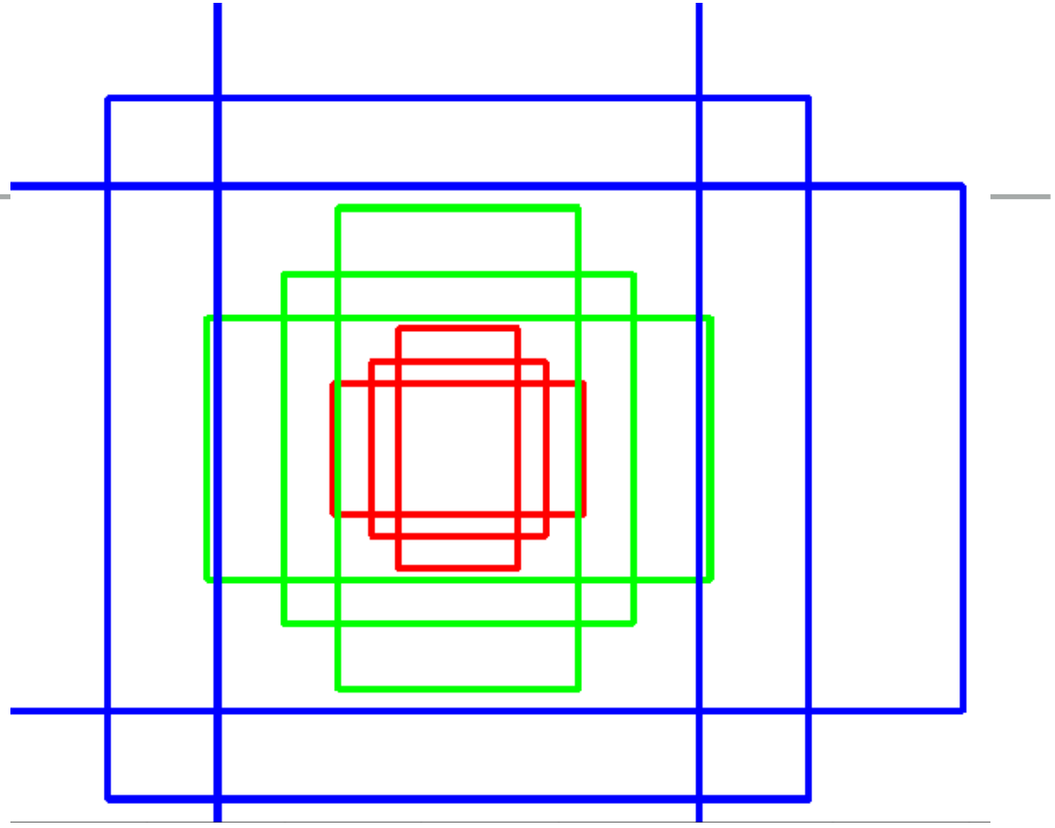
https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/

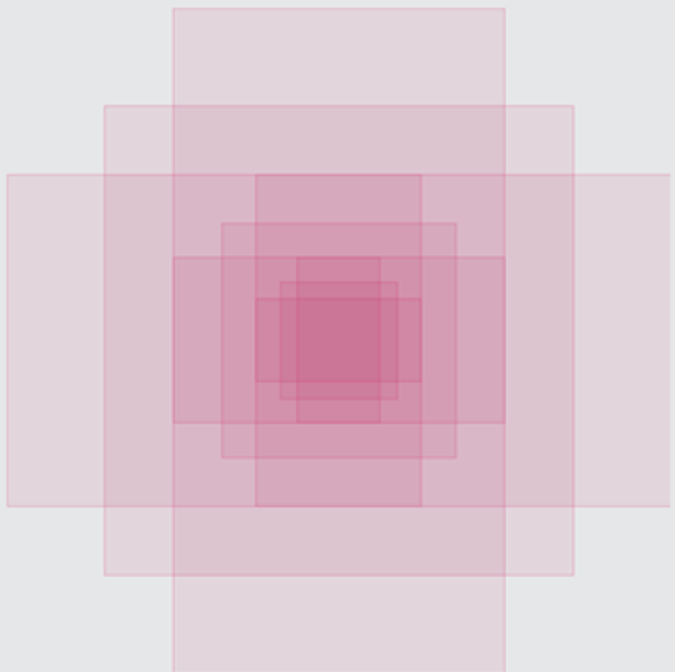
REGION PROPOSAL NETWORK

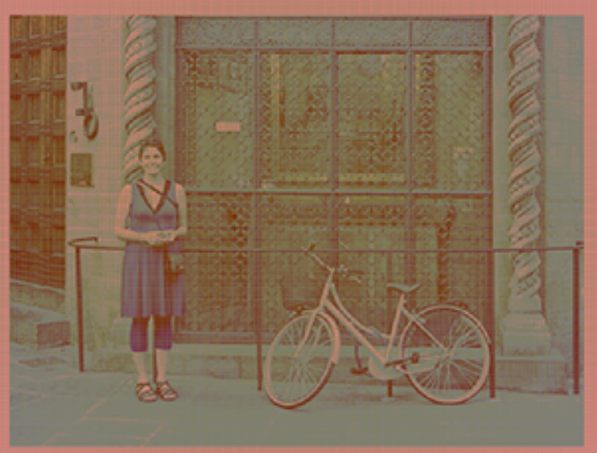
- "anchors": grid of default multi scale-detections
- Training minibatch: select random pos + neg regions (pos bias!)
- Multi task loss: L_cls + L_bbox
- Further stages and testing: select top N detections.



REGION PROPOSAL NETWORK







https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/

TRAINING THE REGION PROPOSAL NETWORK

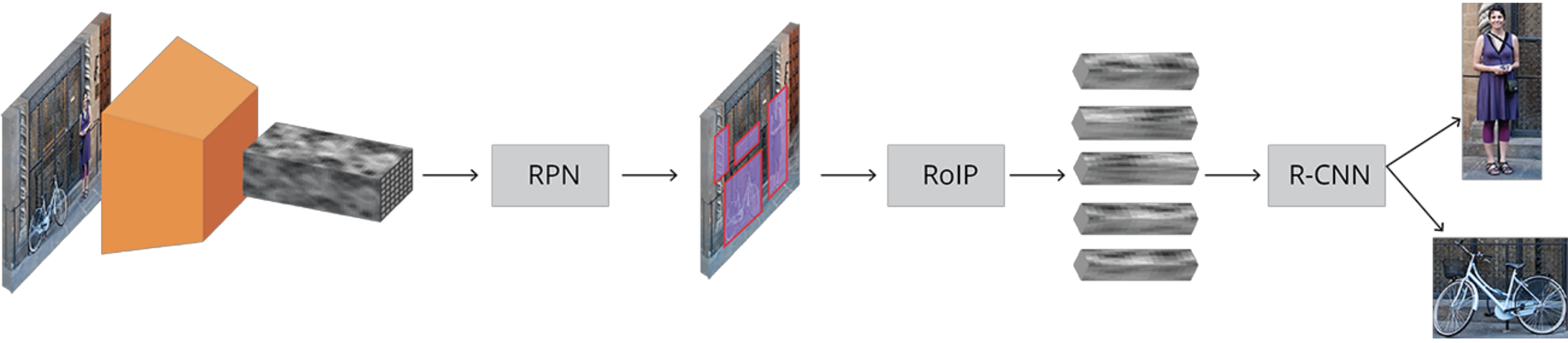
- *We assign a positive label … the anchor/anchors with the highest Intersection- over-Union (IoU) overlap with a ground-truth box, or an anchor that has an IoU overlap higher than 0.7 with any ground-truth box. We assign a negative label to a non-positive anchor if its IoU ratio is lower than 0.3 for all ground-truth boxes. Anchors that are neither positive nor negative do not contribute to the training objective.*

- *we randomly sample 256 anchors in an image to compute the loss function of a mini-batch, where the sampled positive and negative anchors have a ratio of up to 1:1. If there are fewer than 128 positive samples in an image, we pad the mini-batch with negative ones.*

- They trained first the RPN then the Faster R-CNN, but now **joint training** dominates

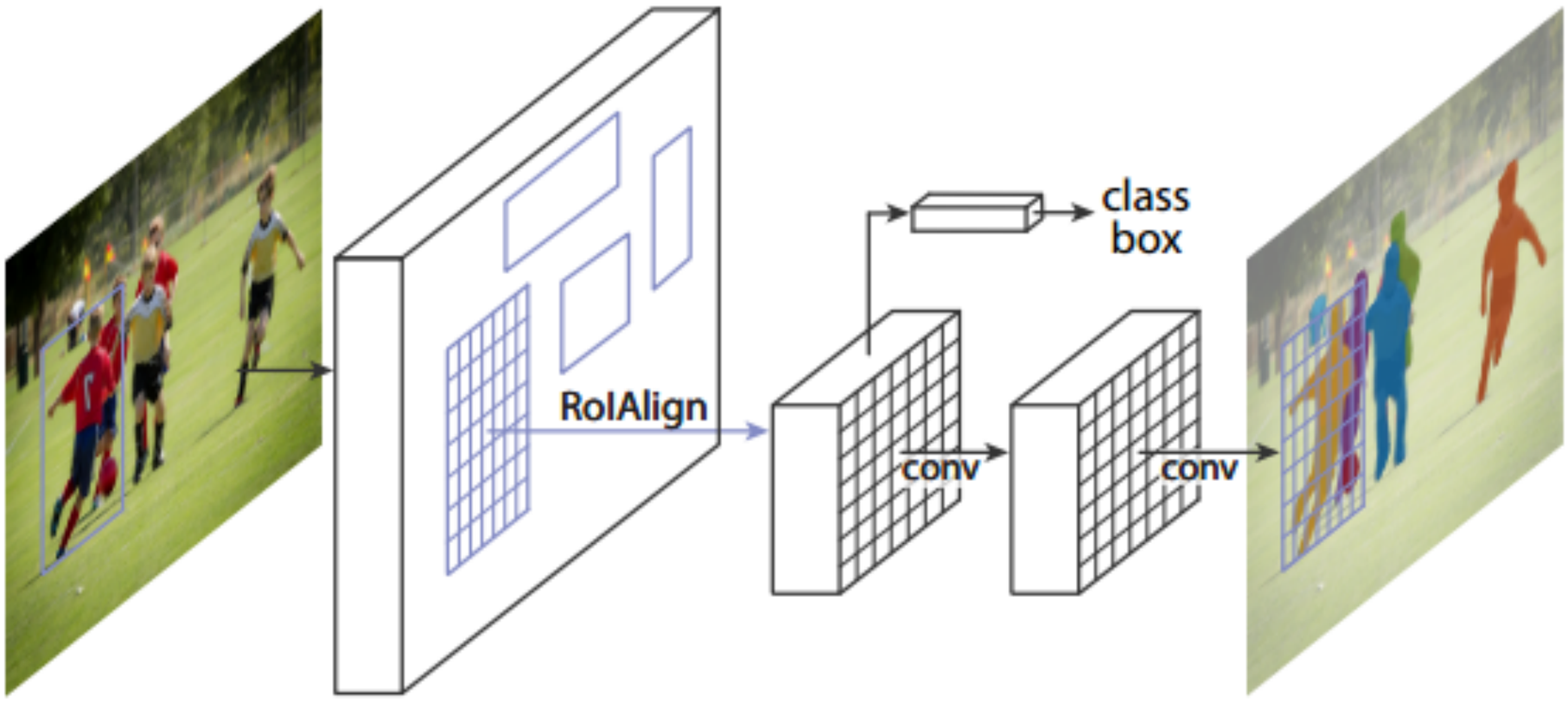
FASTER R-CNN

- Fast R-CNN + CNN generated region proposals
- Down to 5fps then even faster now



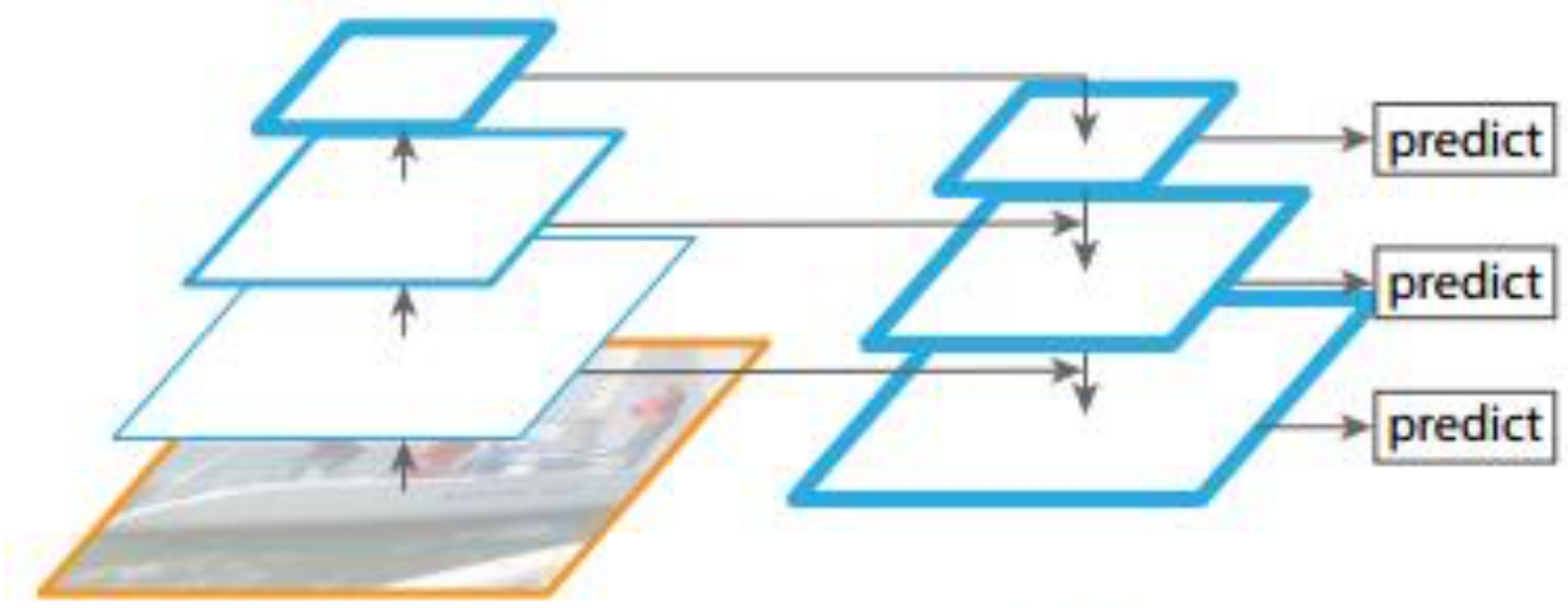
FOR INSTANCE SEGMENTATION: MASK R-CNN

- He, Kaiming, et al. "Mask r-cnn." 2017 (~2000 citations)
- Binary mask per ROI + ROI align instead of ROI Pooling, more accurate position matching



FEATURE PYRAMID NETWORK

- Lin, Tsung-Yi, et al. "Feature pyramid networks for object detection." 2017. (~1000 citations)
- Higher resolution high semantics layers for accurate small object detection



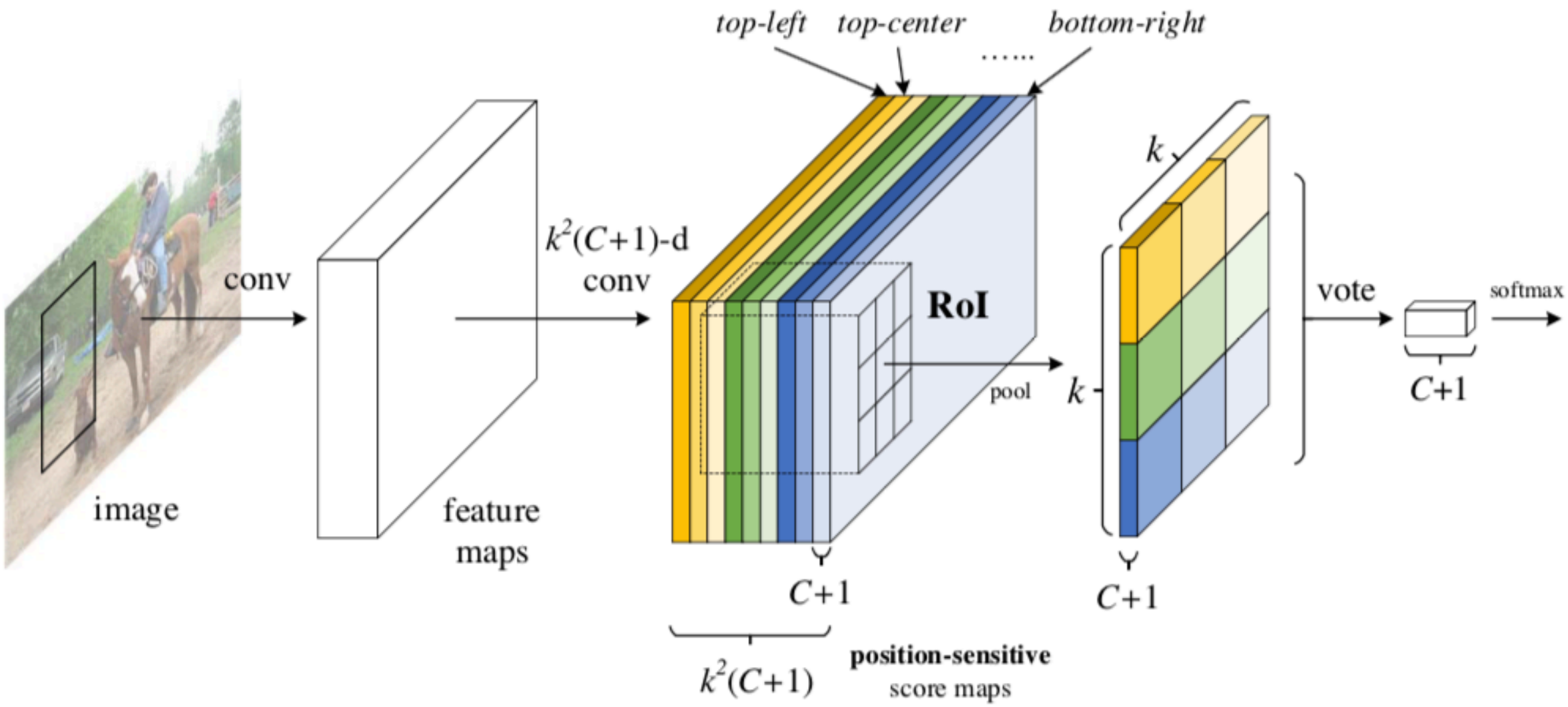
(d) Feature Pyramid Network

COMPETITIVE ALTERNATIVE TO FASTER R-CNN

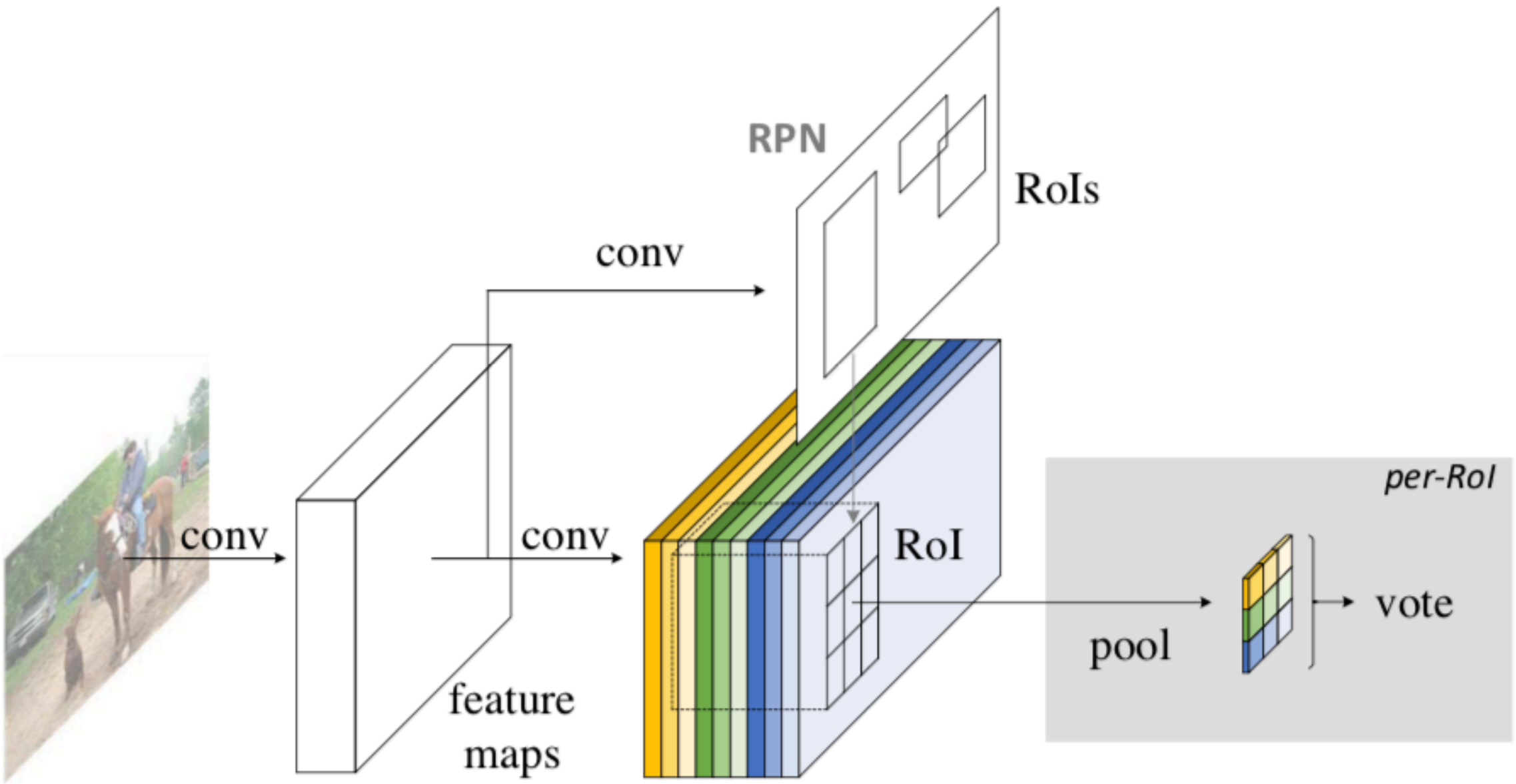
R-FCN: OBJECT DETECTION VIA REGION-BASED FULLY CONVOLUTIONAL NETWORKS

R-FCN

- Dai, Jifeng, et al. "R-fcn: Object detection via region-based fully convolutional networks." 2016.

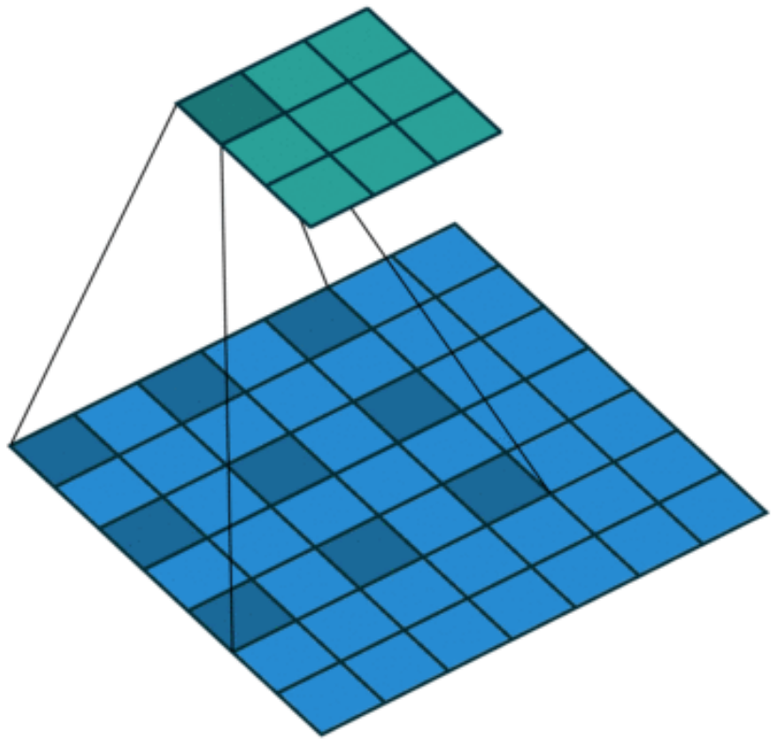


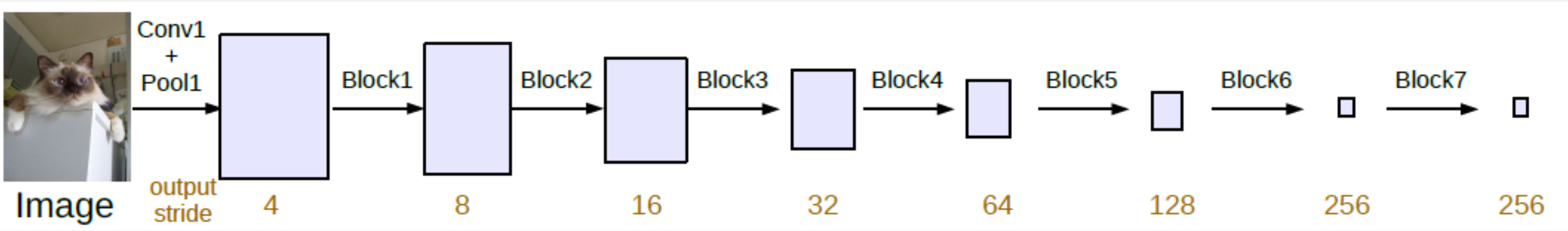
R-FCN



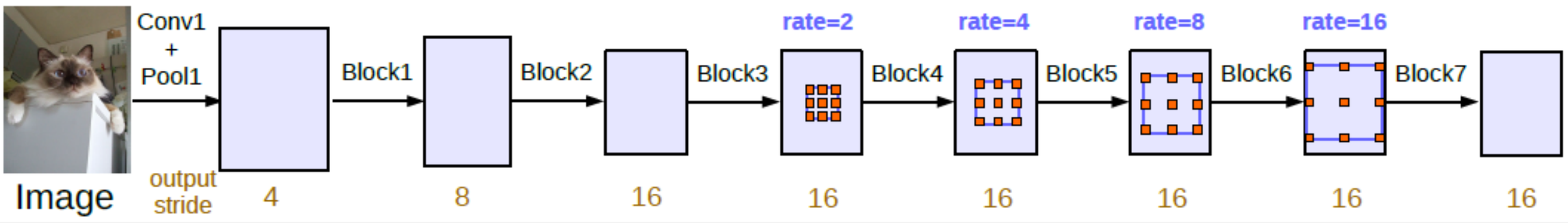
ATROUS, DLIATED CONVOLUTION TRICK

- Trick for increasing the resolution (decreasing the stride) of deep features





(a) Going deeper without atrous convolution.

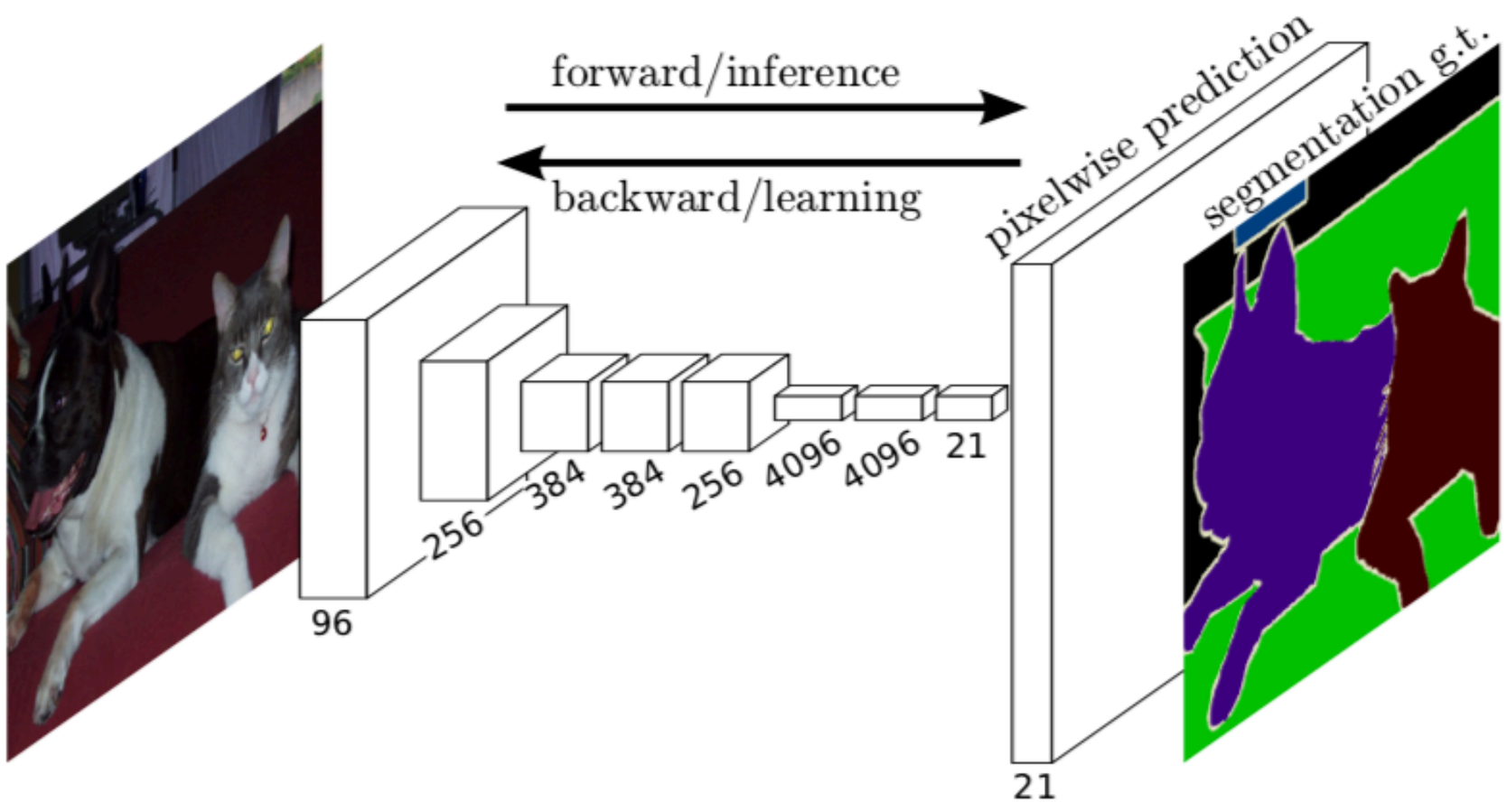


(b) Going deeper with atrous convolution. Atrous convolution with $rate > 1$ is applied after block3 when $output\_stride = 16$.

SEMANTIC SEGMENTATION WITH CONVOLUTIONAL NEURAL NETWORKS

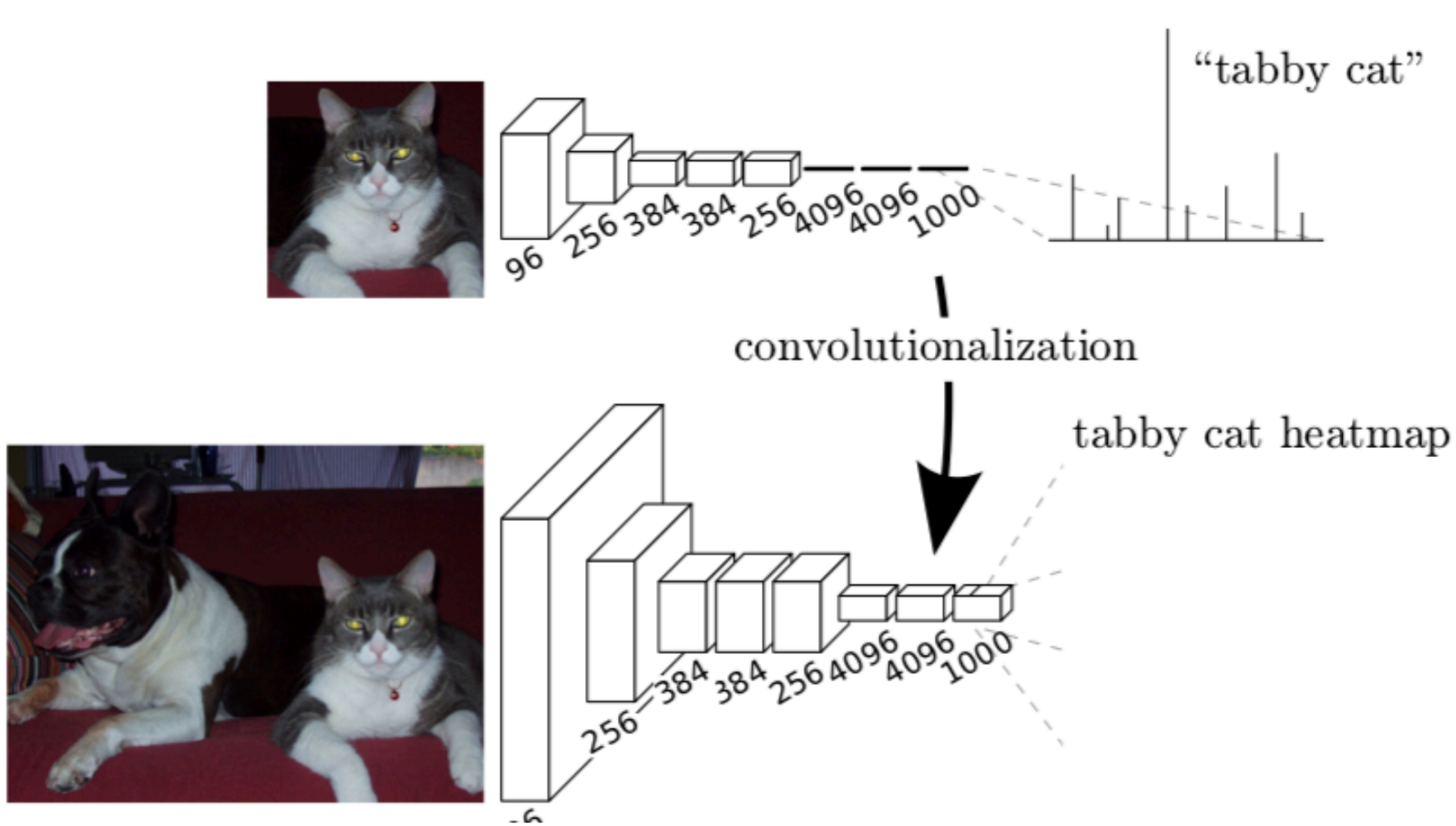
FULLY CONVOLUTIONAL NETWORKS FOR SEMANTIC SEGMENTATION

- Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." 2015. (~8000 citations)



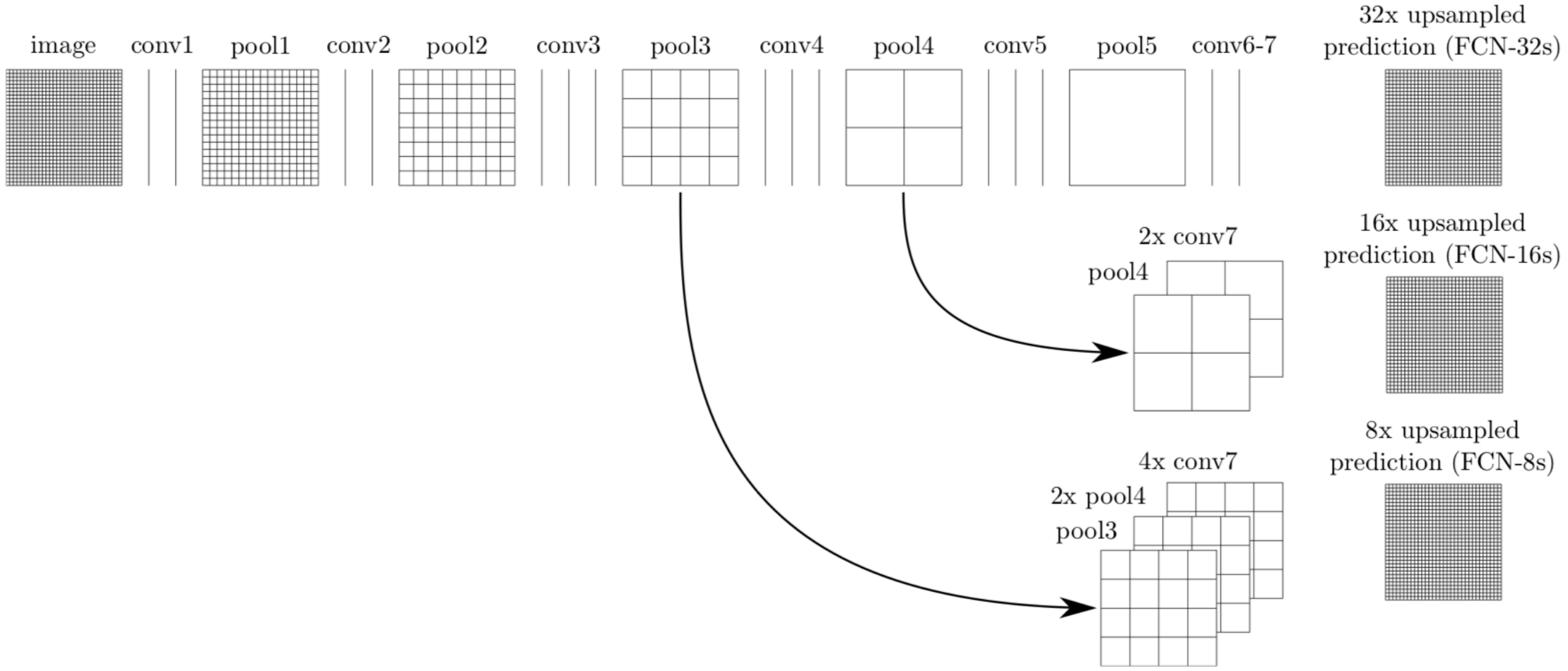
FULLY CONVOLUTIONAL NETWORKS FOR SEMANTIC SEGMENTATION

- Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." 2015.



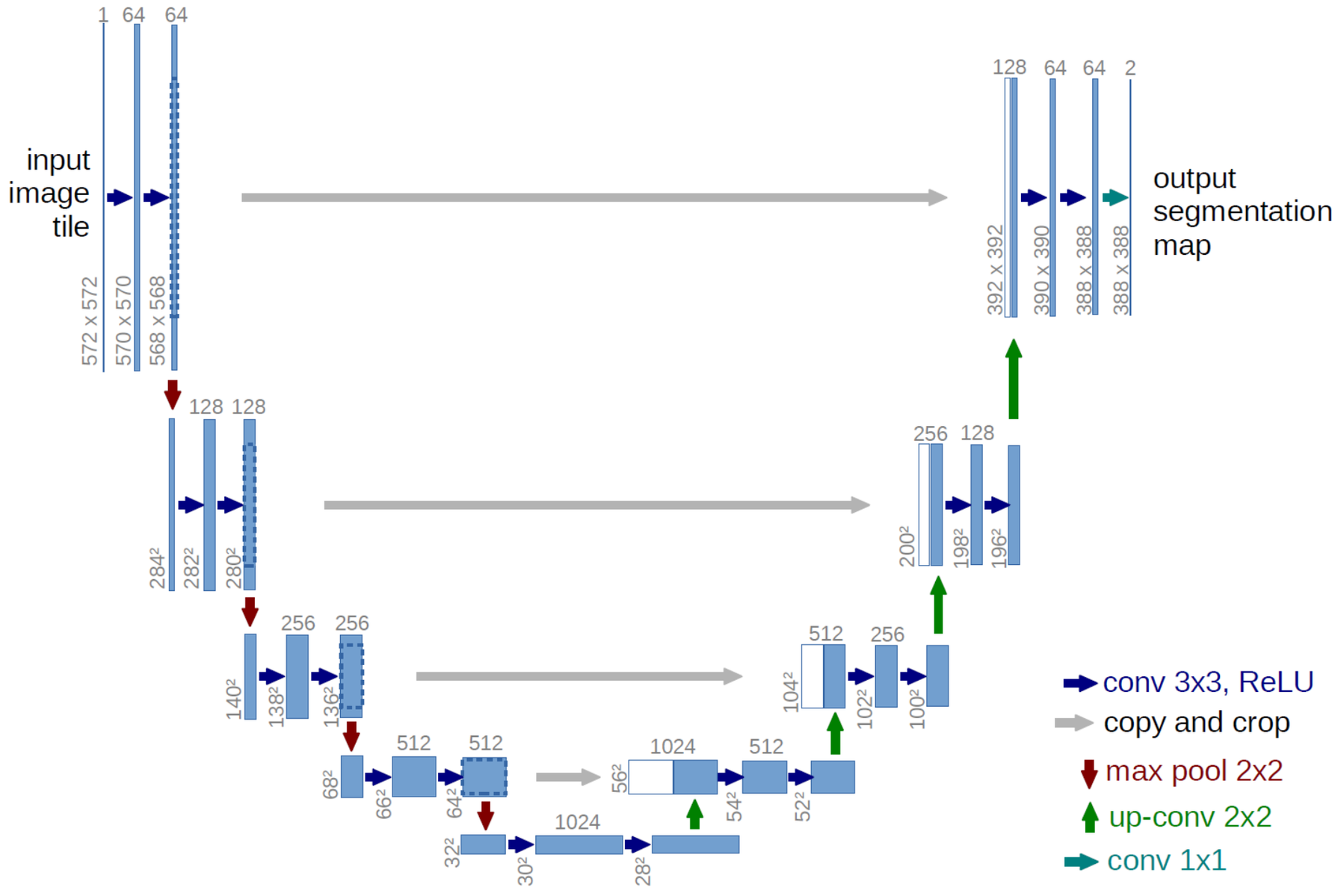
FULLY CONVOLUTIONAL NETWORKS FOR SEMANTIC SEGMENTATION

- Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." 2015.



U-NET

- Ronneberger, Olaf, et al. "U-net: Convolutional networks for biomedical image segmentation. " 2015. (~5000 citations)

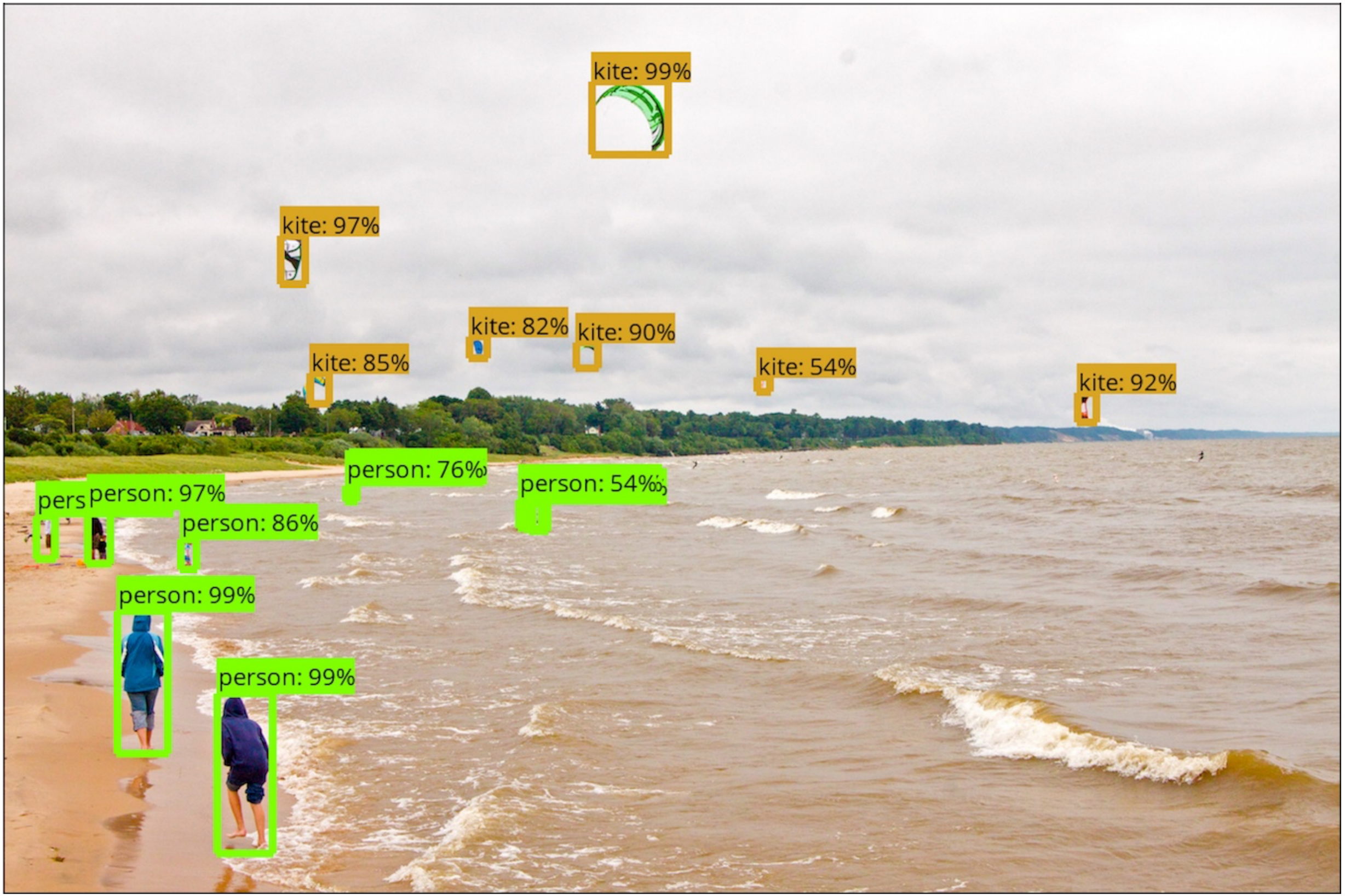


STATE OF THE ART SOLUTIONS
IMPLICITLY DEAL WITH CONTEXT …

WHAT IS THAT?

CONTEXT HELPS WITH SMALL OBJECTS. IT GIVES YOU A STRONG PRIOR PROBABILITY!

COCO: COMMON OBJECTS IN "CONTEXT"


kite: 99%
kite: 97%
kite: 82%
kite: 90%
kite: 85%
kite: 54%
kite: 92%
person: 76%
person: 54%
pers
person: 97%
person: 86%
person: 99%
person: 99%