國立高雄科技大學
National Kaohsiung University of Science and Technology

Region Proposal Network (PRN)

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun

Neural Information Processing Systems, 2015

Speaker: Shih-Shinh Huang

March 30, 2020

S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Network", *NIPS*, 2015

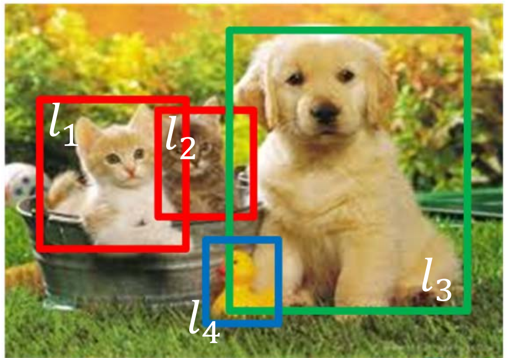
Outline

- Introduction
  - About Object Detection
  - RPN Background
- RPN Architecture
  - Anchors
  - Convolutional Networks
  - Contribution
- RPN Training
  - Training Steps
  - Anchor Selection
  - Loss Function Definition

Introduction

- About Object Detection
  - Input:
    - I: input image
    - $\{c_1, c_2, \ldots, c_n\}$: object classes to be detected
  - Output:
    - $\{r_1, r_2, \ldots r_m\}$: bounding boxes of m detected objects
    - $\{l_1, l_2, \ldots l_m\}$: class labels of all detected objects

$C = \{\textbf{cat}, \textbf{dog}, \textbf{duck}\}$



$l_1 = \textbf{cat}$

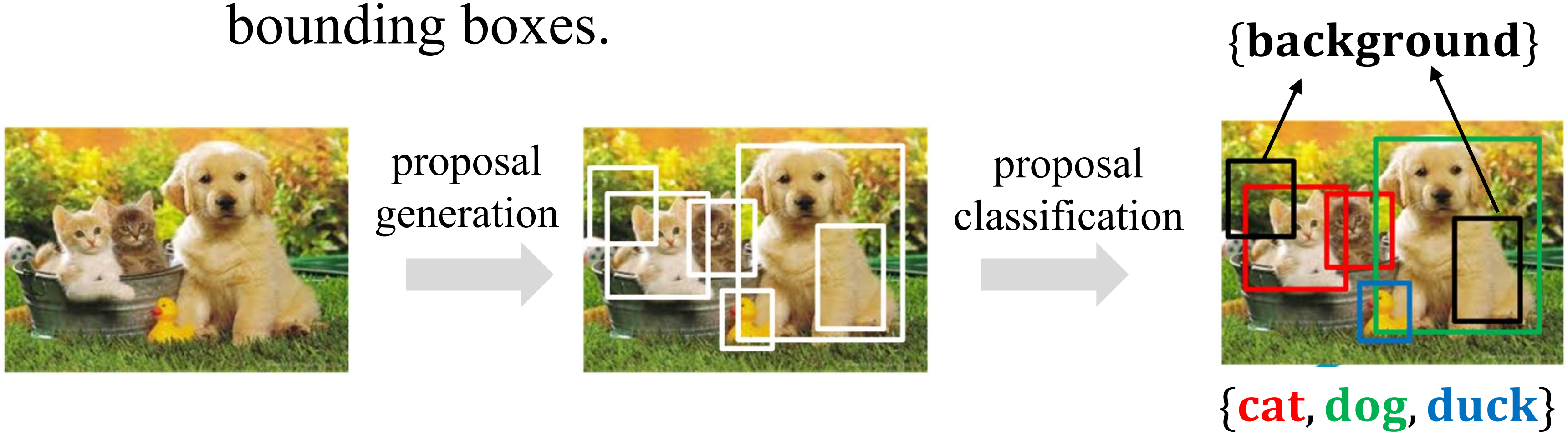
$l_2 = \textbf{cat}$

$l_3 = \textbf{dog}$

$l_4 = \textbf{duck}$

Introduction

- About Object Detection
  - Two stages of an object detector
    - proposal generation: capture the rectangular bounding boxes of all possible objects
    - proposal classification: assign class labels to all bounding boxes.



Introduction

- RPN Background: R-CNN
  - Region-based CNN (R-CNN) has made remarkable advances in object detection

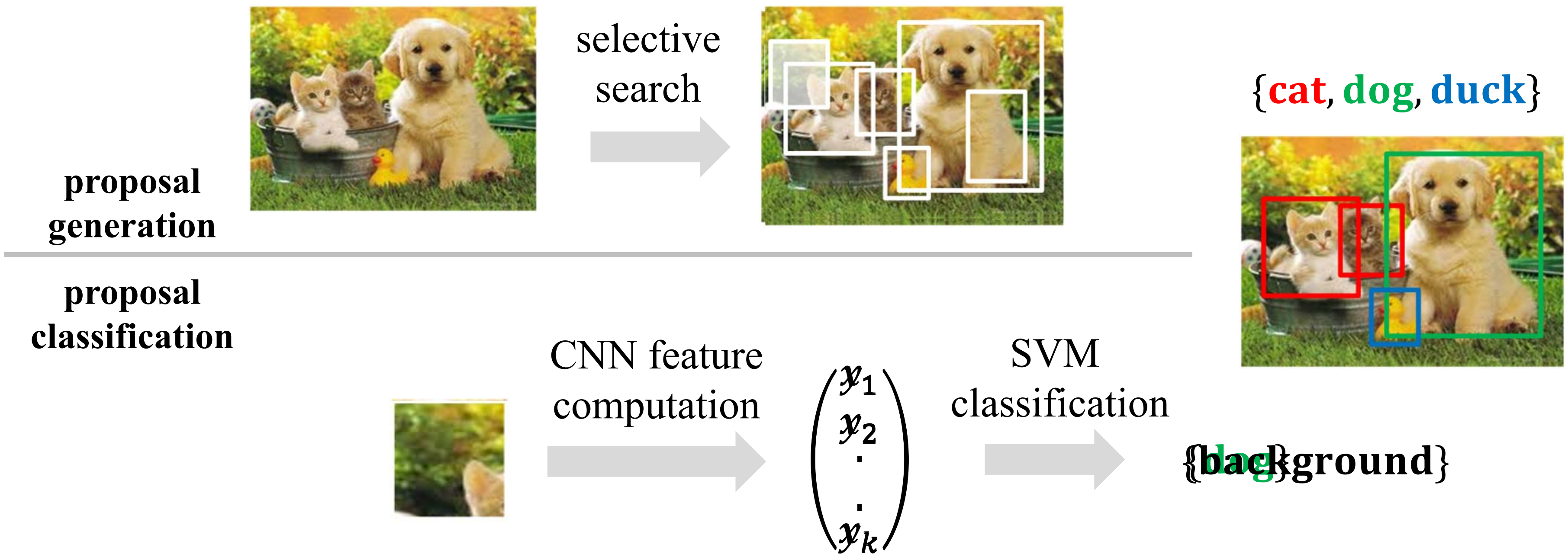
R. Girshick, et. al. “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” *CVPR*, 2014

- use selective search to generate object proposals
- compute CNN features of each proposal and classify it by support vector machines (SVMs).

J. R. Uijlings et. al. “Selective Search for Object Detection” *IJCV*, 2013.

Introduction

- RPN Background: R-CNN

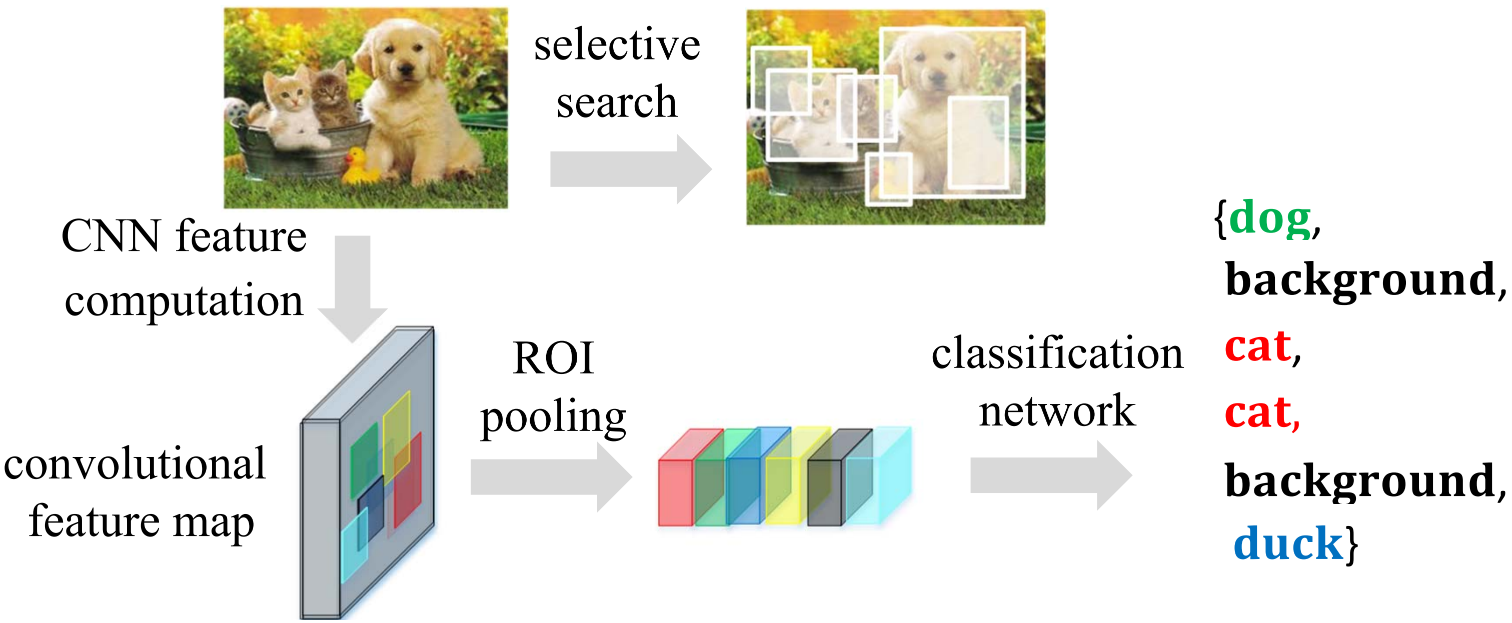


drawback: R-CNN is slow

reason: it performs convolution forward pass for each proposal

Introduction

- RPN Background: Fast R-CNN
  - share convolution feature map across proposals



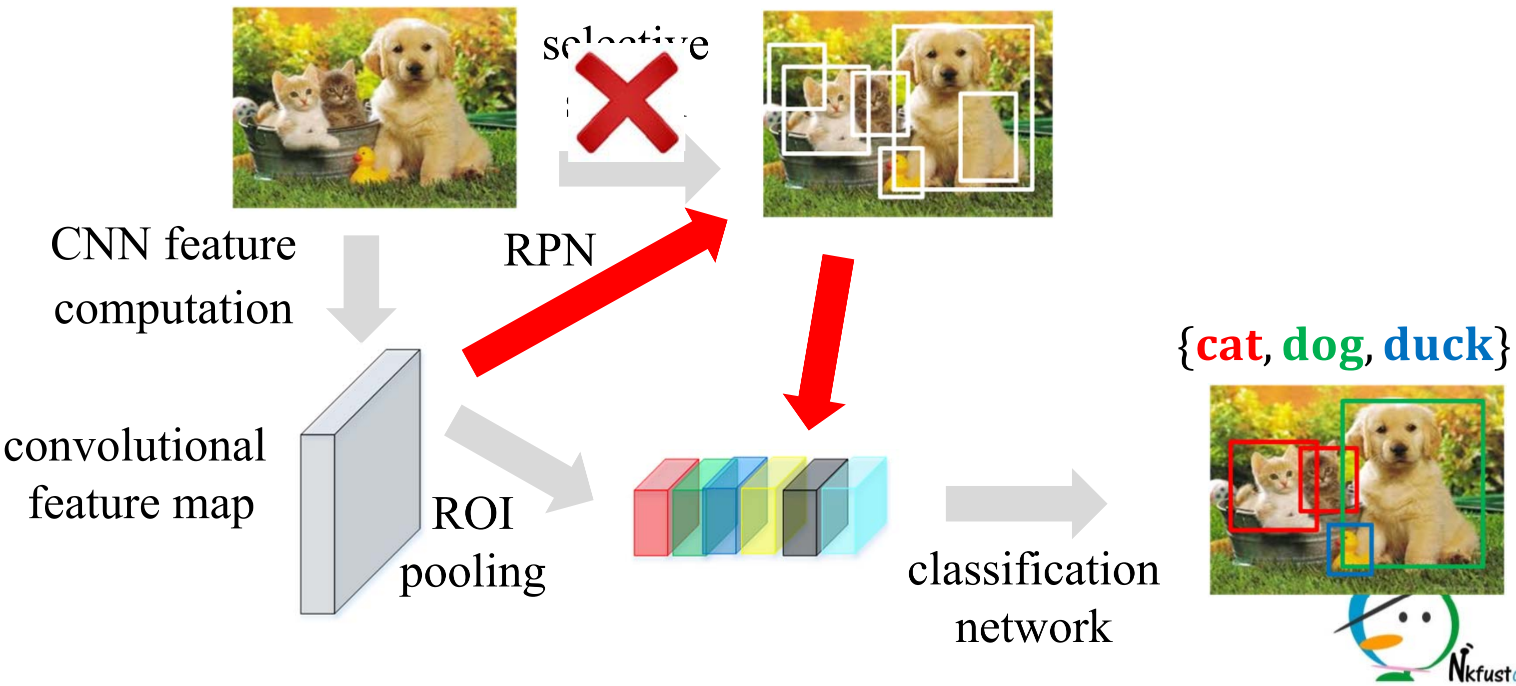
selective search becomes computational bottleneck

R. Girshick, “Fast R-CNN,” *CVPR*, 2015

Introduction

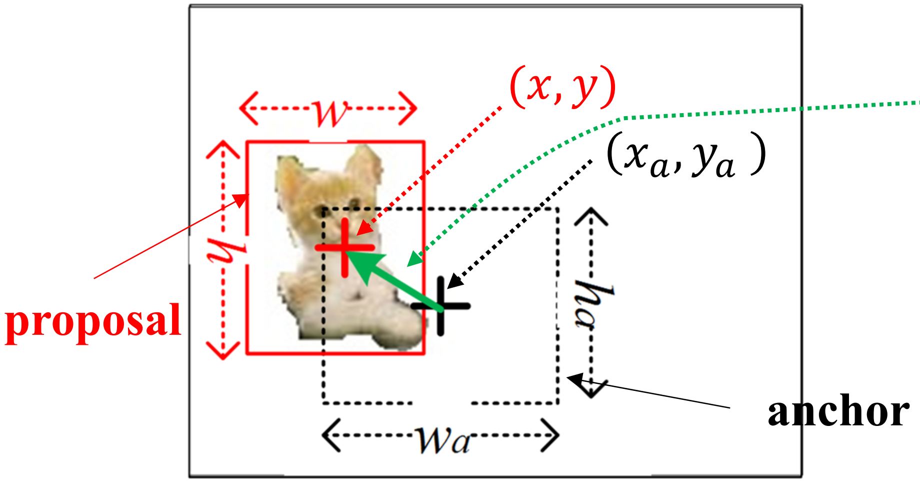
reason: misses sharing with clas

- RPN Background: Faster R-CNN
  - generate proposals based on convolutional feature map shared with classification stage



RPN Architecture

- Anchors: proposal parameterization
  - Anchors are reference boxes for encoding proposal
  - A proposal is parameterized as (d_x, d_y, d_w, d_h) relative to an anchor .



relative encoding

$$d_x = (x - x_a)/w_a$$

$$d_y = (y - y_a)/h_a$$

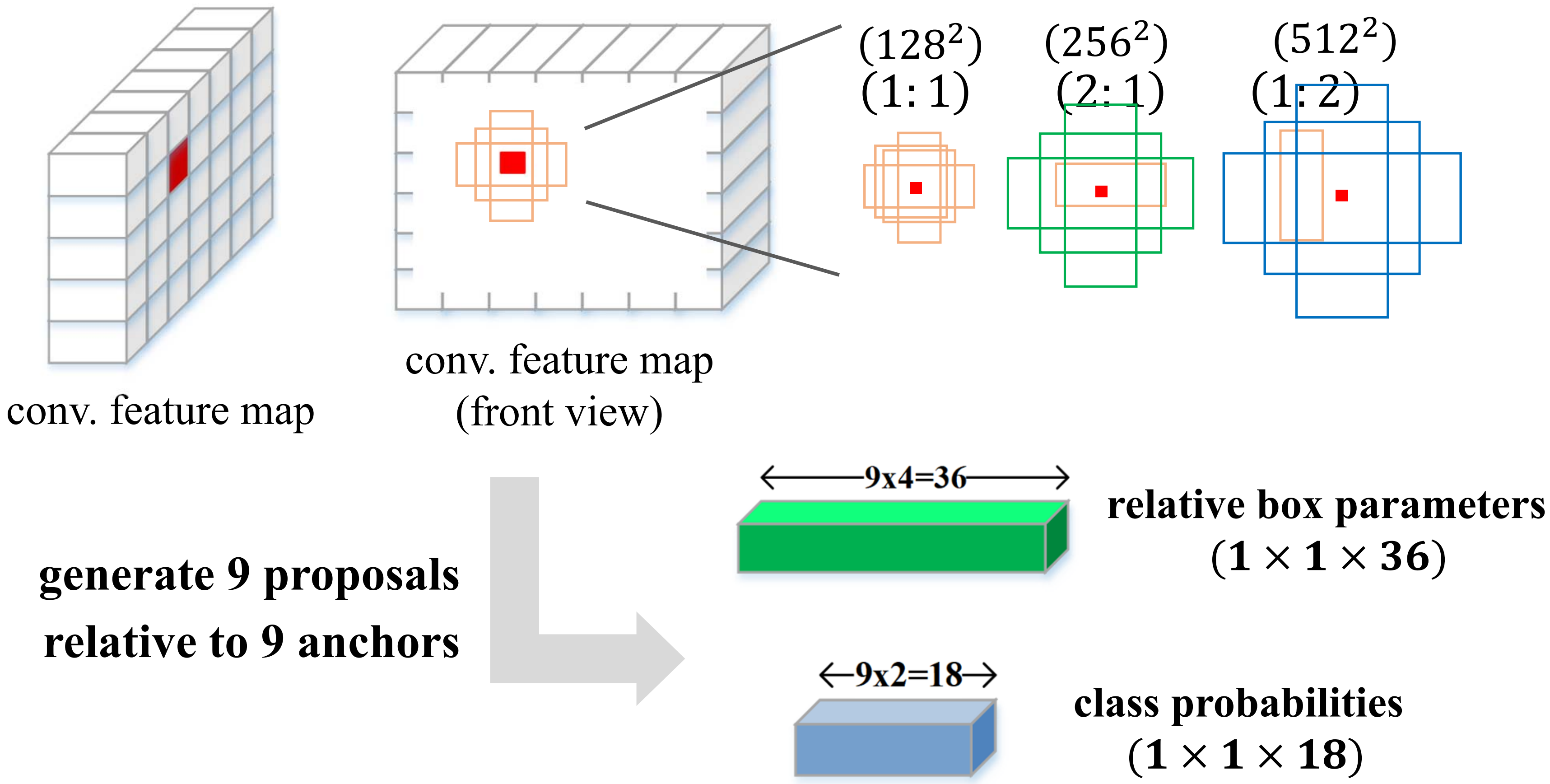
$$d_w = w/w_a$$

$$d_h = h/h_a$$

RPN Architecture

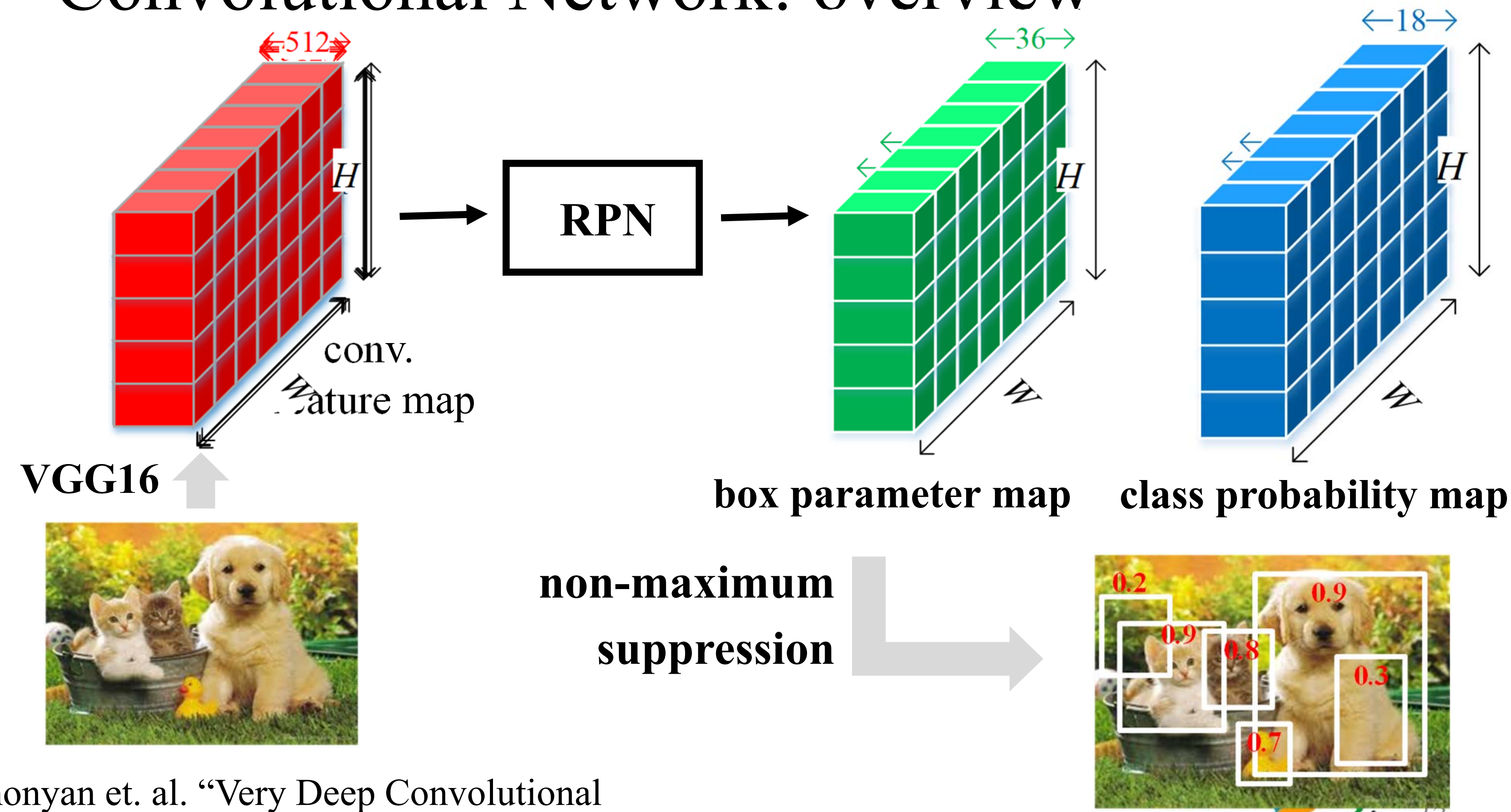
- Anchors: proposal generation
  - attach 9 anchors centered at each point of the conv. feature map
    - 3 aspect ratios for detecting various object types
    - 3 scales for dealing with scaling variance
  - predict one proposal with 6 parameters with respect to each anchor
    - (d_x, d_y, d_w, d_h): relative box parameters
    - (p_{obj}, p_{bg}): object/background class probabilities

RPN Architecture



RPN Architecture

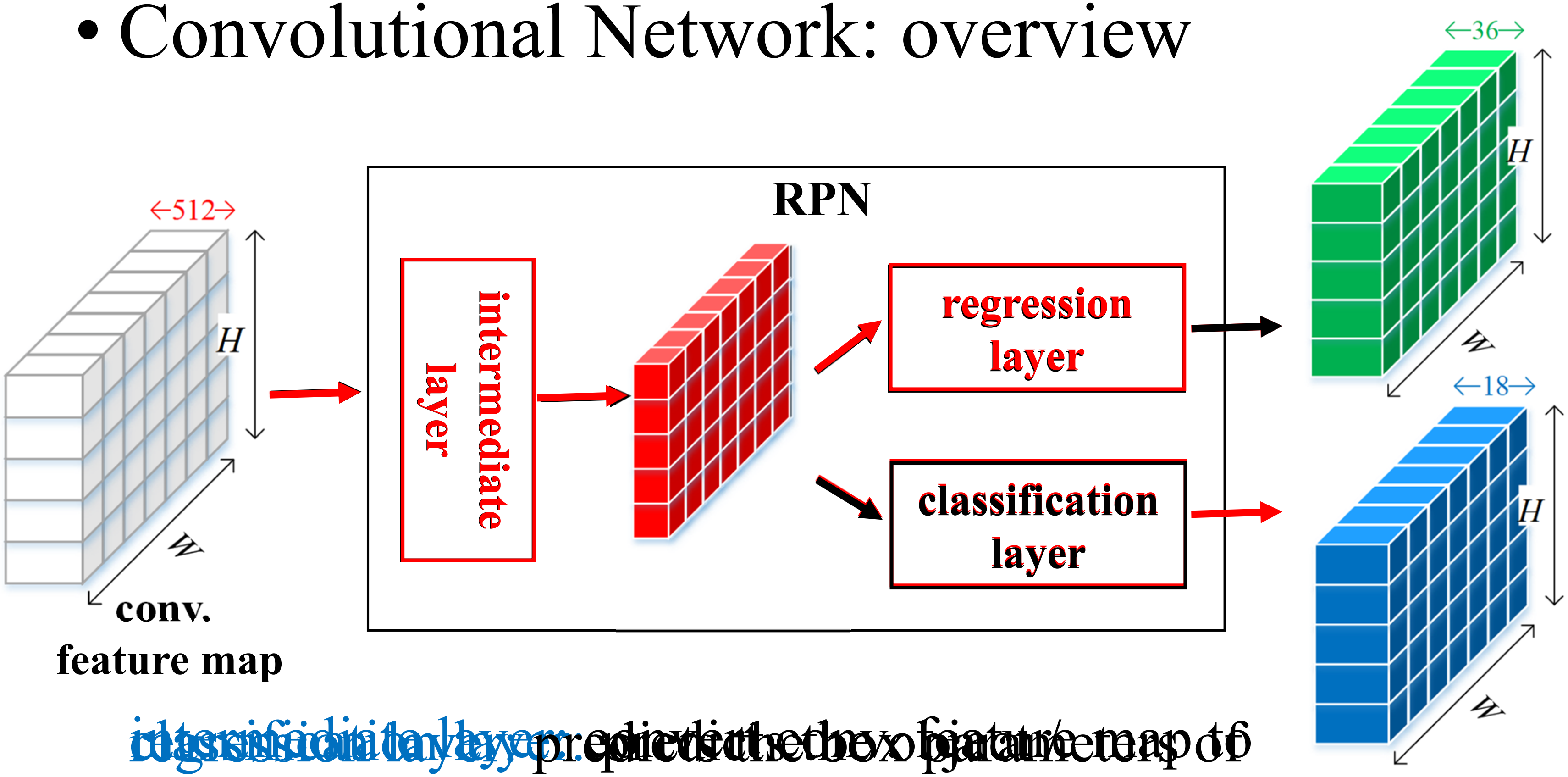
- Convolutional Network: overview



K. Simonyan et. al. “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *ICLR*, 2015.

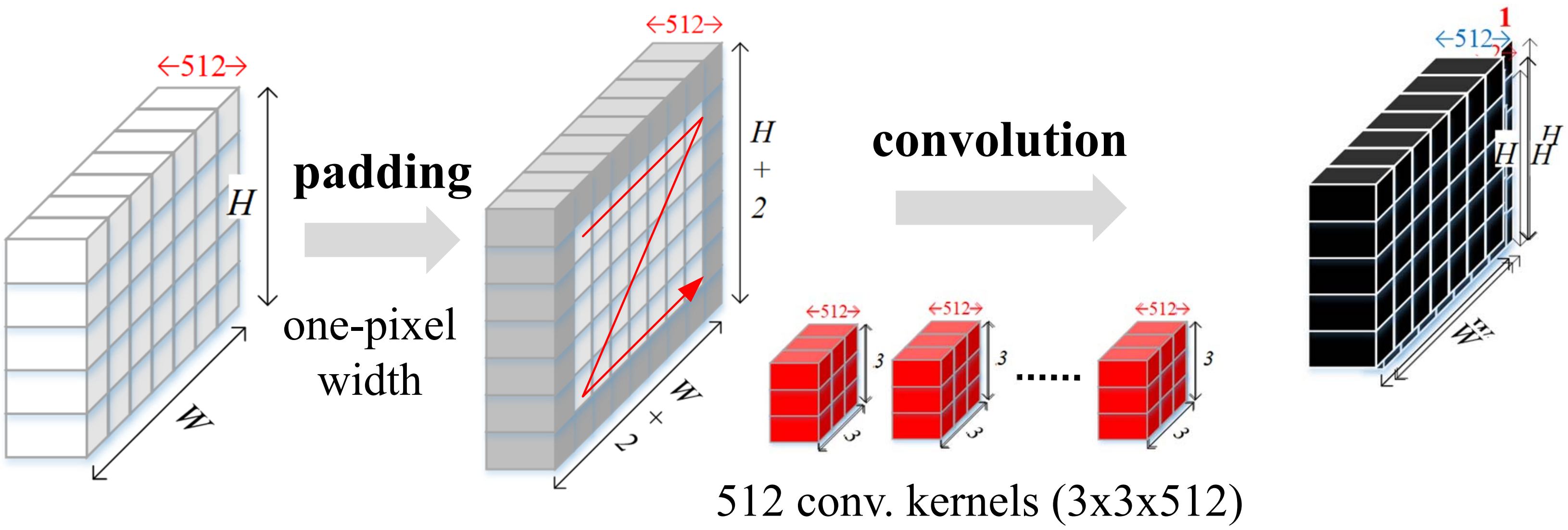
RPN Architecture

- Convolutional Network: overview



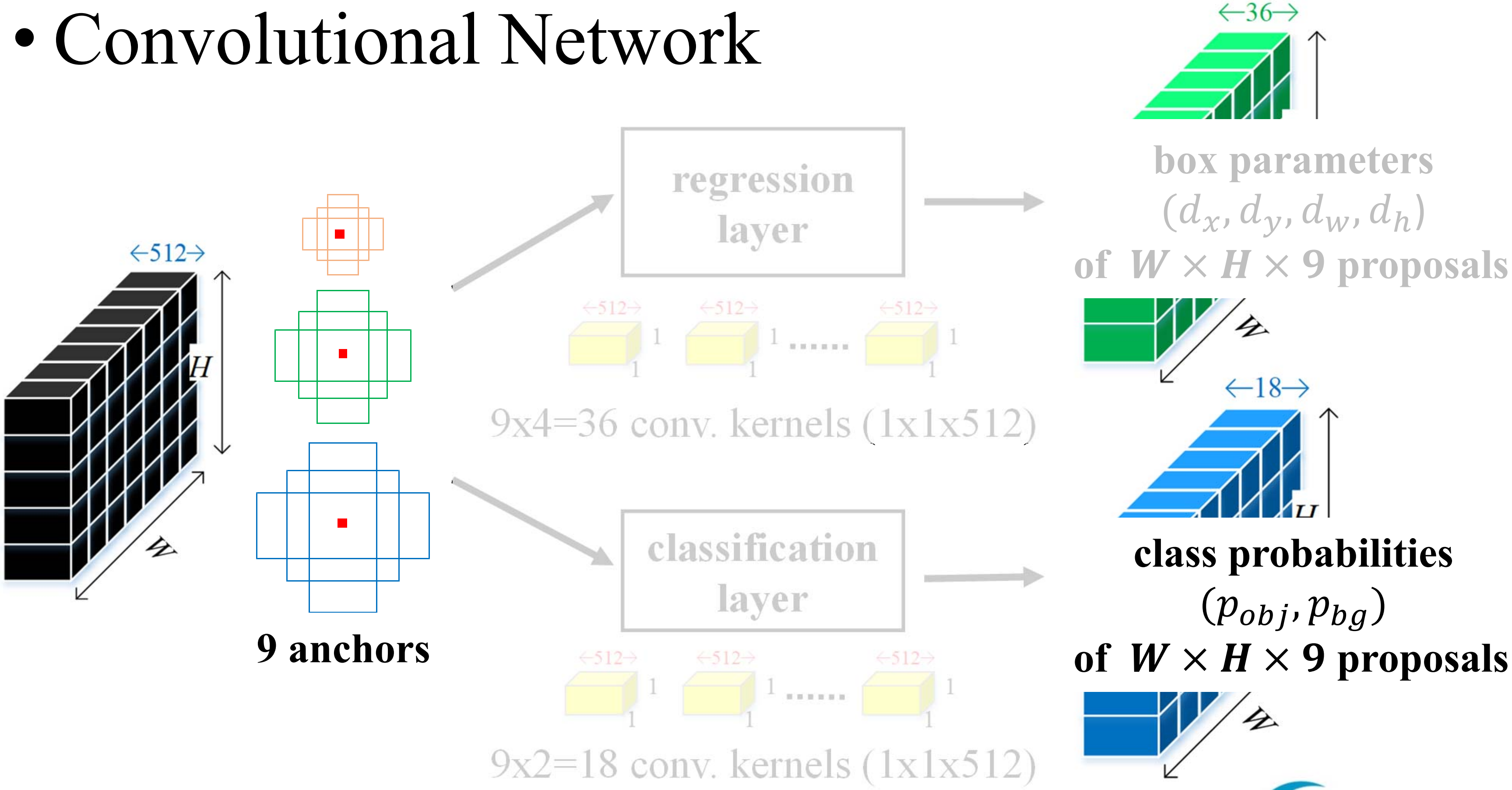
RPN Architecture

- Convolutional Network
  - intermediate layer: extract feature map for proposal generation.



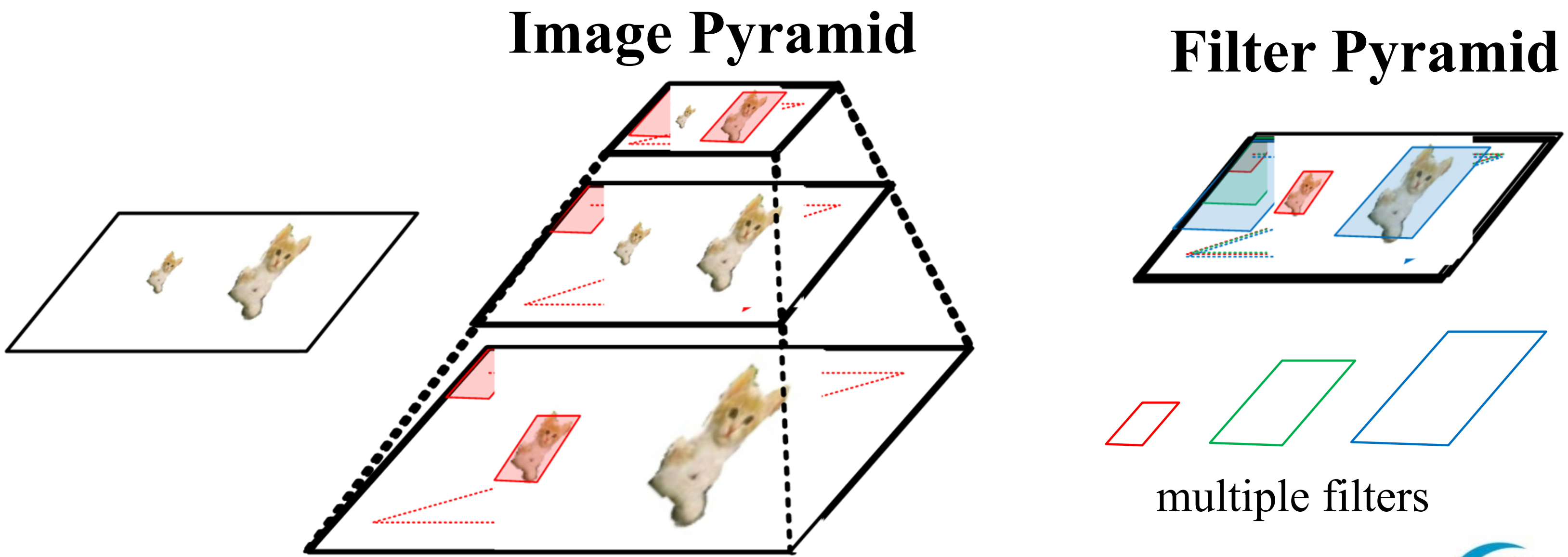
RPN Architecture

- Convolutional Network



RPN Architecture

- Contribution: Multi-Scale Anchors
  - There are two basic approaches for detecting objects in multiple scales.

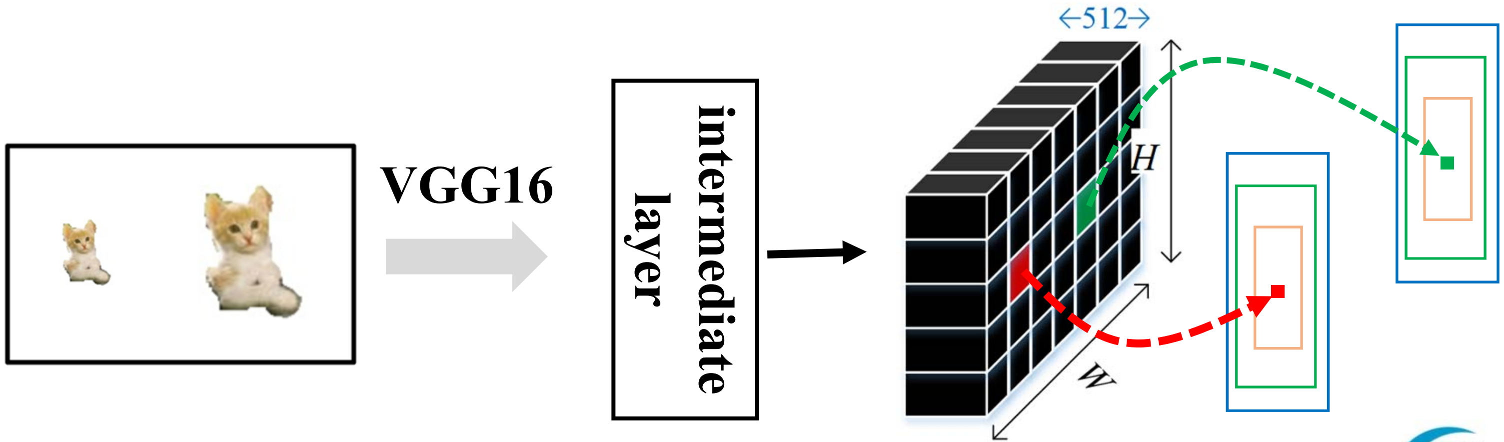


Both approaches are time-consuming

RPN Architecture

- Contribution: Multi-Scale Anchors
  - RPN localizes objects in multiple scales by multi-scale anchors (anchor pyramid)

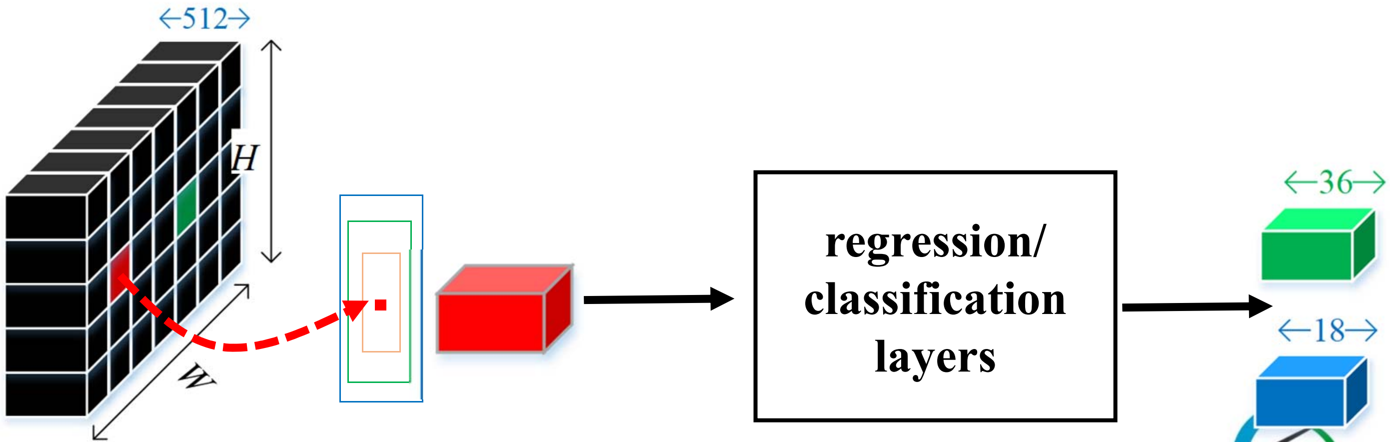
➡ relies on the image of single scale



RPN Architecture

- Contribution: Multi-Scale Anchors
  - Multi-scale anchors centered at the same point share the same feature.

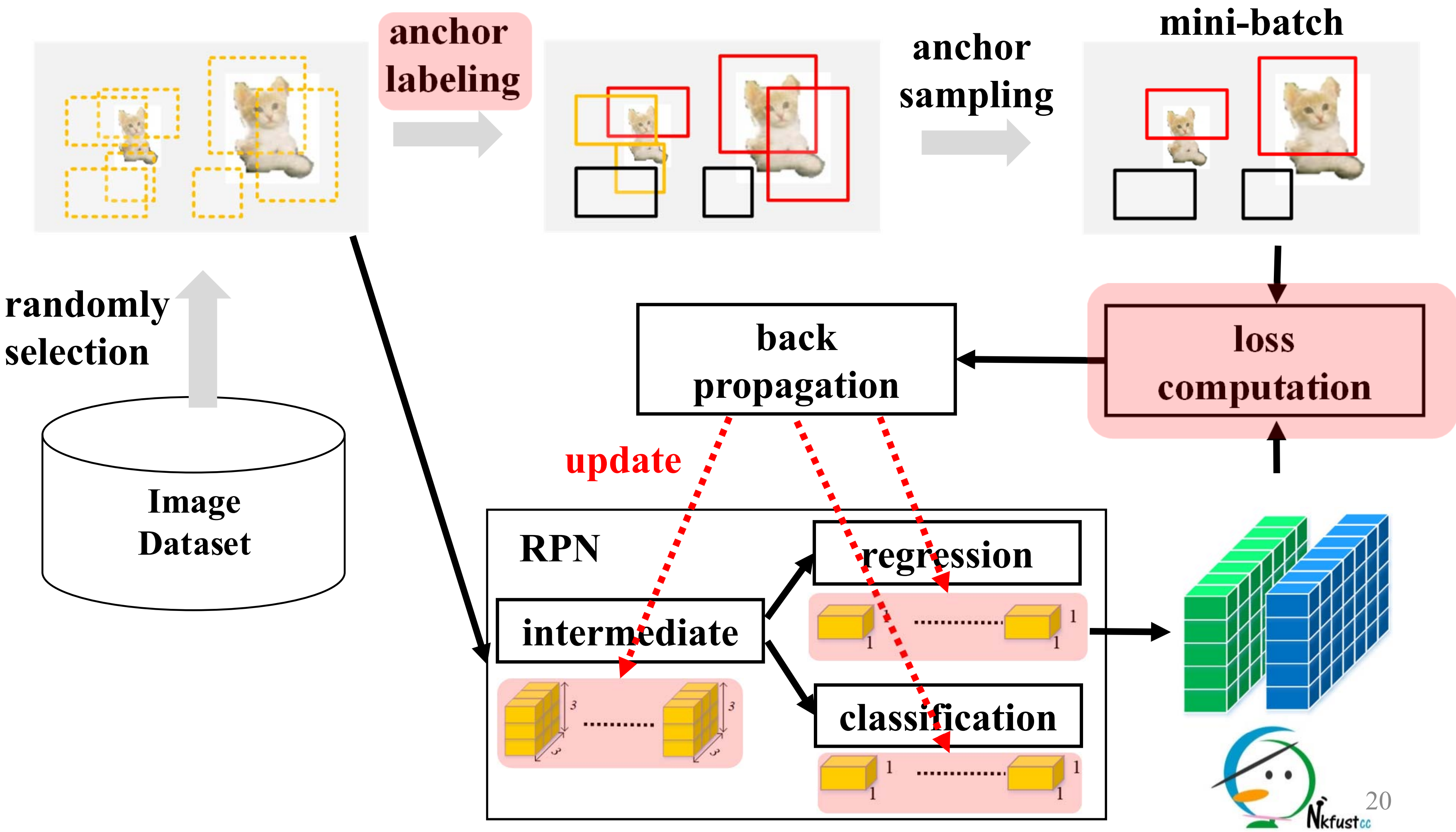
➡ addresses scaling variance without extra cost



RPN Training

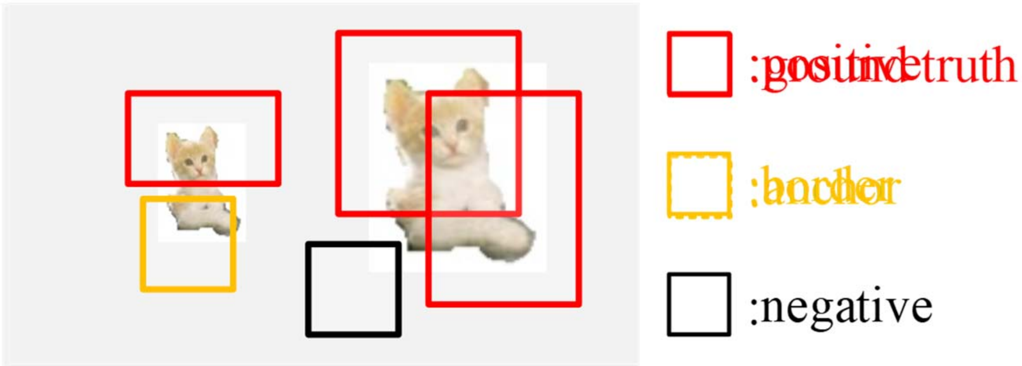
- Training Step
  - select an image from training dataset
  - assign a label to each anchor
  - form a mini-batch consisting of 256 anchors
    - 128 positive (object) anchors
    - 128 negative (background) anchors
  - minimize the defined loss function
    - optimization: stochastic gradient descent (SGD)
    - learning rate: 0.001(first 60k); 0.0001 (next 20k)

RPN Training

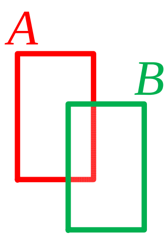


RPN Training

- Anchor Labeling
  - assign a class label to each anchor

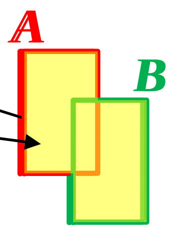


  - use IoU for measuring box overlap



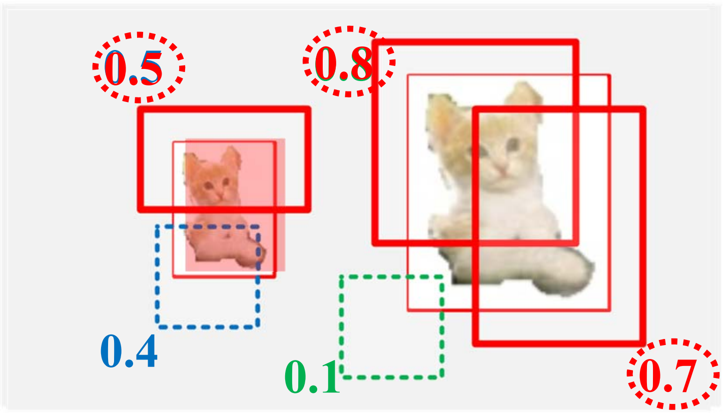
$$\mathbf{IoU}(\boldsymbol{A},\boldsymbol{B}) = \frac{\boldsymbol{A} \cap \boldsymbol{B}}{\boldsymbol{A} \cup \boldsymbol{B}}$$

Intersection over Union



RPN Training

- Anchor Labeling
  - positive anchor labeling
    - has the highest IoU with a ground-truth box
    - has IoU ≥ 0.7 with any ground-truth box



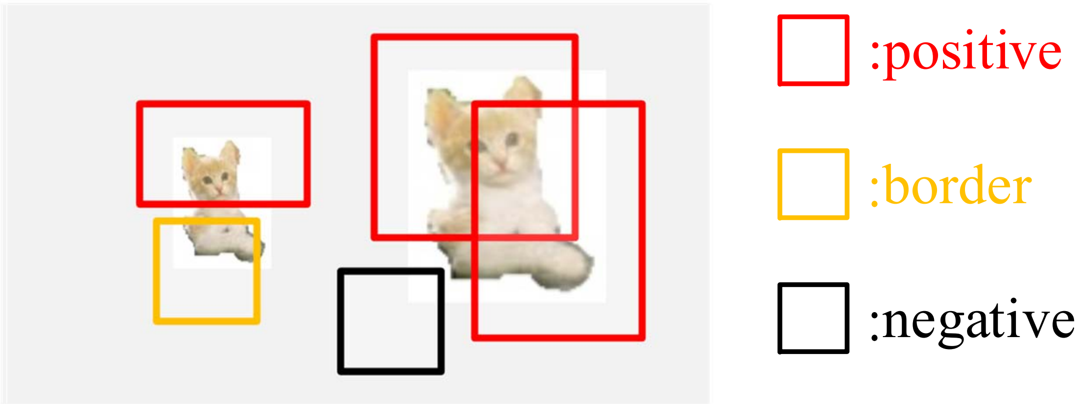
:ground truth :positive

:anchor

:negative

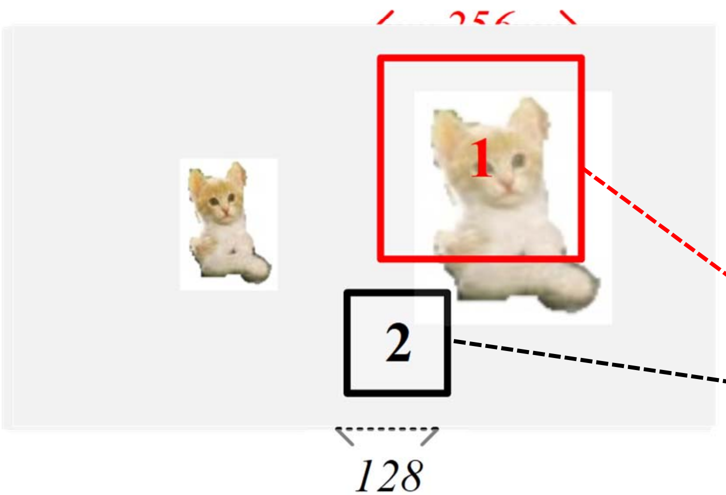
RPN Training

- Anchor Labeling
  - negative anchor labeling
    - has IoU $<$ 0.3 for all ground-truth boxes
  - border anchor labeling
    - neither positive nor negative



RPN Training

- Loss Function $L(.)$
  - let $D = \{\hat{d}^{(i)}, \hat{p}^{(i)}\}_{i=1}^{256}$ be the selected anchors
    - $\hat{p}^{(i)} = (\hat{p}_{obj}^{(i)}, \hat{p}_{bg}^{(i)})$: class probabilities of ith anchor
    - $\hat{d}^{(i)} = (\hat{d}_x^{(i)}, \hat{d}_y^{(i)}, \hat{d}_w^{(i)}, \hat{d}_h^{(i)})$: box parameters of ground-truth box associated with ith anchor



$$\hat{d}^{(1)} = \left(\frac{32}{256}, \frac{32}{256}, \frac{256}{256}, \frac{282}{256}\right)$$

$$\hat{d}^{(1)} = (0.125, 0.125, 1.0, 1.1)$$

$$\hat{d}^{(2)} = \left(\frac{64}{256}, \frac{64}{256}, \frac{256}{128}, \frac{282}{128}\right)$$

$$\hat{d}^{(2)} = (0.25, 0.25, 2.0, 2.2)$$

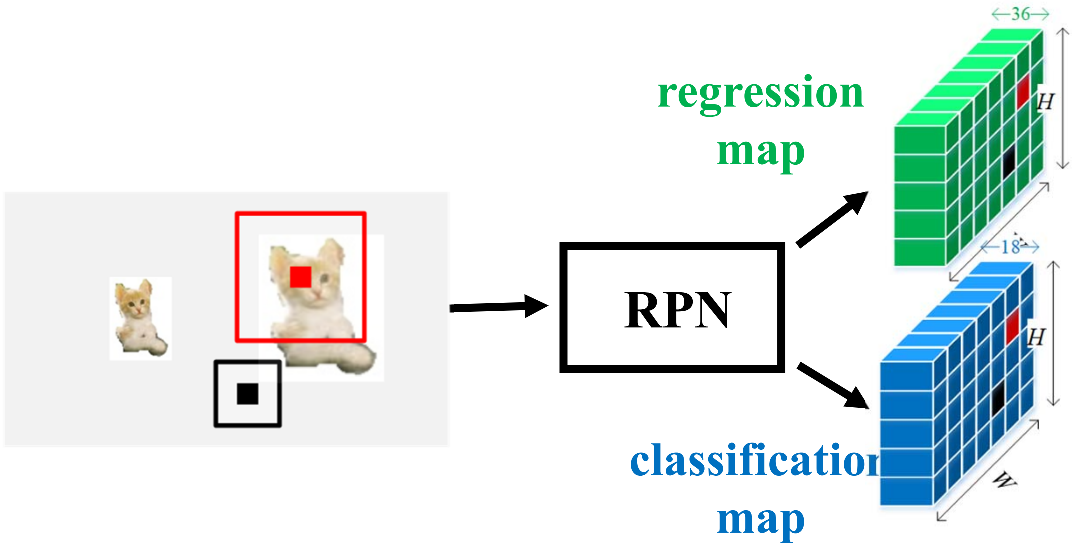
$$\hat{p}^{(1)} = (\mathbf{1.0}, 0.0)$$

$$\hat{p}^{(2)} = (0.0, \mathbf{1.0})$$

RPN Training

- Loss Function $L(.)$
  - $\{d^{(i)}, p^{(i)}\}$: proposal parameters predicted by RPN via anchors
  - $L(.)$ evaluates fitness of $\{d^{(i)}, p^{(i)}\}$ to $\{\hat{d}^{(i)}, \hat{p}^{(i)}\}$

$$L(\{\boldsymbol{d}^{(i)}, \boldsymbol{p}^{(i)}\}) = \underbrace{\sum_i \hat{\boldsymbol{p}}_{obj}^{(i)} \times L_{reg}(\boldsymbol{d}^{(i)}, \hat{\boldsymbol{d}}^{(i)})}_{\text{regression term}} + \underbrace{\sum_i L_{cls}(\boldsymbol{p}^{(i)}, \hat{\boldsymbol{p}}^{(i)})}_{\text{classification term}}$$



$d^{(1)} = (0.4, 0.2, 1.2, 1.2)$

$d^{(2)} = (0.2, 0.35, 2.0, 2.0)$

$p^{(1)} = (\mathbf{0.6}, 0.4)$

$p^{(2)} = (0.2, \mathbf{0.8})$

RPN Training

- Loss Function $L(.)$: $\sum_i \hat{\boldsymbol{p}}_{obj}^{(i)} \times L_{reg}\left(\boldsymbol{d}^{(i)}, \hat{\boldsymbol{d}}^{(i)}\right)$
  - $\hat{\boldsymbol{p}}_{obj}^{(i)}$:only depends on positive anchors
  - L_{reg} : evaluate difference via smooth L_1 function

$$L_{reg}\left(\boldsymbol{d}^{(i)}, \hat{\boldsymbol{d}}^{(i)}\right) = \boldsymbol{L_1}\left(\boldsymbol{d}_x^{(i)} - \hat{\boldsymbol{d}}_x^{(i)}\right) + \boldsymbol{L_1}\left(\boldsymbol{d}_y^{(i)} - \hat{\boldsymbol{d}}_y^{(i)}\right)$$

$$+\boldsymbol{L_1}\left(\log\left(\boldsymbol{d}_w^{(i)}\right) - \log\left(\hat{\boldsymbol{d}}_w^{(i)}\right)\right) + \boldsymbol{L_1}\left(\log\left(\boldsymbol{d}_h^{(i)}\right) - \log\left(\hat{\boldsymbol{d}}_h^{(i)}\right)\right)$$

	d_x	d_y	d_w	d_h
$d^{(1)}=$	(0.4	0.2	1.2	1.2)
$\hat{d}^{(1)}=$	(0.125	0.125	1.0	1.1)

$$= \boldsymbol{L_1}(\boldsymbol{0.4} - 0.125) + \boldsymbol{L_1}(\boldsymbol{0.2} - \boldsymbol{0.125})$$

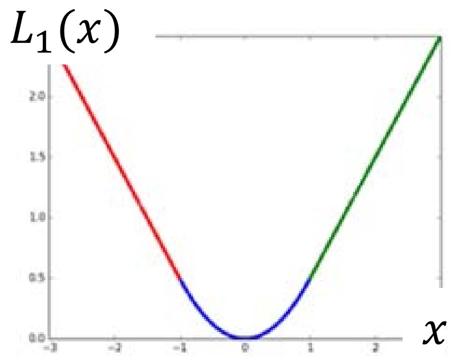
$$+\boldsymbol{L_1}\left(\log(\boldsymbol{1.2}) - \log(\boldsymbol{1.0})\right)$$

$$+\boldsymbol{L_1}\left(\log(\boldsymbol{1.2}) - \log(\boldsymbol{1.1})\right)$$

RPN Training

- Loss Function $L(.)$: $\sum_i \hat{\boldsymbol{p}}_{\boldsymbol{obj}}^{(\boldsymbol{i})} \times L_{reg}(\boldsymbol{d}^{(\boldsymbol{i})}, \hat{\boldsymbol{d}}^{(\boldsymbol{i})})$

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



$$= \boldsymbol{L_1}(0.275) + \boldsymbol{L_1}(0.075) + \boldsymbol{L_1}(0.18) + \boldsymbol{L_1}(0.09)$$

$$= (0.5 \times (0.275)^2) + (0.5 \times (0.005)^2)$$
$$+ (0.5 \times (0.18)^2) + (0.5 \times (0.09)^2)$$

RPN Training

- Loss Function $L(.)$: $\sum_i L_{cls}(\boldsymbol{p}^{(i)}, \widehat{\boldsymbol{p}}^{(i)})$
  - $L_{cls}(.)$: log loss function.

$$L_{cls}(\boldsymbol{p}^{(i)}, \widehat{\boldsymbol{p}}^{(i)}) = -\left(\widehat{\boldsymbol{p}}_{obj}^{(i)} \times \log\left(\boldsymbol{p}_{obj}^{(i)}\right) + \widehat{\boldsymbol{p}}_{bg}^{(i)} \times \log\left(\boldsymbol{p}_{bg}^{(i)}\right)\right)$$

$$\begin{array}{lcc} & \boldsymbol{p}_{obj} & \boldsymbol{p}_{bg} \\ \boldsymbol{p}^{(1)} = & (\ 0.6 & 0.4\ ) \\ \widehat{\boldsymbol{p}}^{(1)} = & (\ 1.0 & 0.0\ ) \\ \boldsymbol{p}^{(2)} = & (\ 0.2 & 0.8\ ) \\ \widehat{\boldsymbol{p}}^{(2)} = & (\ 0.0 & 1.0\ ) \end{array}$$

$$\sum_i L_{cls}(\boldsymbol{p}^{(i)}, \widehat{\boldsymbol{p}}^{(i)})$$

$$= -\left(\widehat{\boldsymbol{p}}_{obj}^{(1)} \times \log\left(\boldsymbol{p}_{obj}^{(1)}\right) + \widehat{\boldsymbol{p}}_{bg}^{(1)} \times \log\left(\boldsymbol{p}_{bg}^{(1)}\right)\right)$$

$$= -\left(1.0 \times \log(0.6) + 0.0 \times \log(0.4)\right)$$

$$-\left(\widehat{\boldsymbol{p}}_{obj}^{(2)} \times \log\left(\boldsymbol{p}_{obj}^{(2)}\right) + \widehat{\boldsymbol{p}}_{bg}^{(2)} \times \log\left(\boldsymbol{p}_{bg}^{(2)}\right)\right)$$


thank you
© 1996, 2002 SANRIO CO., LTD.