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

Mask R-CNN

Kaiming He, Georgia Gkioxari, Piotr Dollar,
and Ross Girshick

IEEE Intl. Conf. on Computer Vision (ICCV), 2017

Speaker: Shih-Shinh Huang

March 17, 2022

Outline

- Introduction
  - About Instance Segmentation
  - About Mask R-CNN
- RoI Align
  - Objective
  - Formal Statement
  - Align Steps
- Mask Branch
  - Design Concept
  - Network Architecture
  - Mask Loss

Introduction

- About Instance Segmentation
  - Instance segmentation is a combination of **object detection** and **semantic segmentation**.

object detection
- localize all object instances by bounding boxes
- assign class labels to all object instances

semantic segmentation
- produce the masks of all object instances

Introduction

- About Instance Segmentation
  - Input:
    - I: input image
    - $\{c_1, c_2, \dots, c_n\}$: object classes
  - Output:
    - $\{r_1, r_2, \dots r_k\}$: bounding boxes of k detected objects
    - $\{l_1, l_2, \dots l_k\}$: class labels of all detected objects
    - $\{m_1, m_2, \dots m_k\}$: masks of all 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 Mask R-CNN
  - Mask R-CNN is an extension of Faster R-CNN.

Shaoqing Ren, *et. al.*, "Faster R-CNN Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. on PAMI*, 2017.

- add **a branch** for predicting a segmentation mask to each region of interest (RoI).
- propose **RoI align** to replace RoI pooling for preserving exact spatial locations.

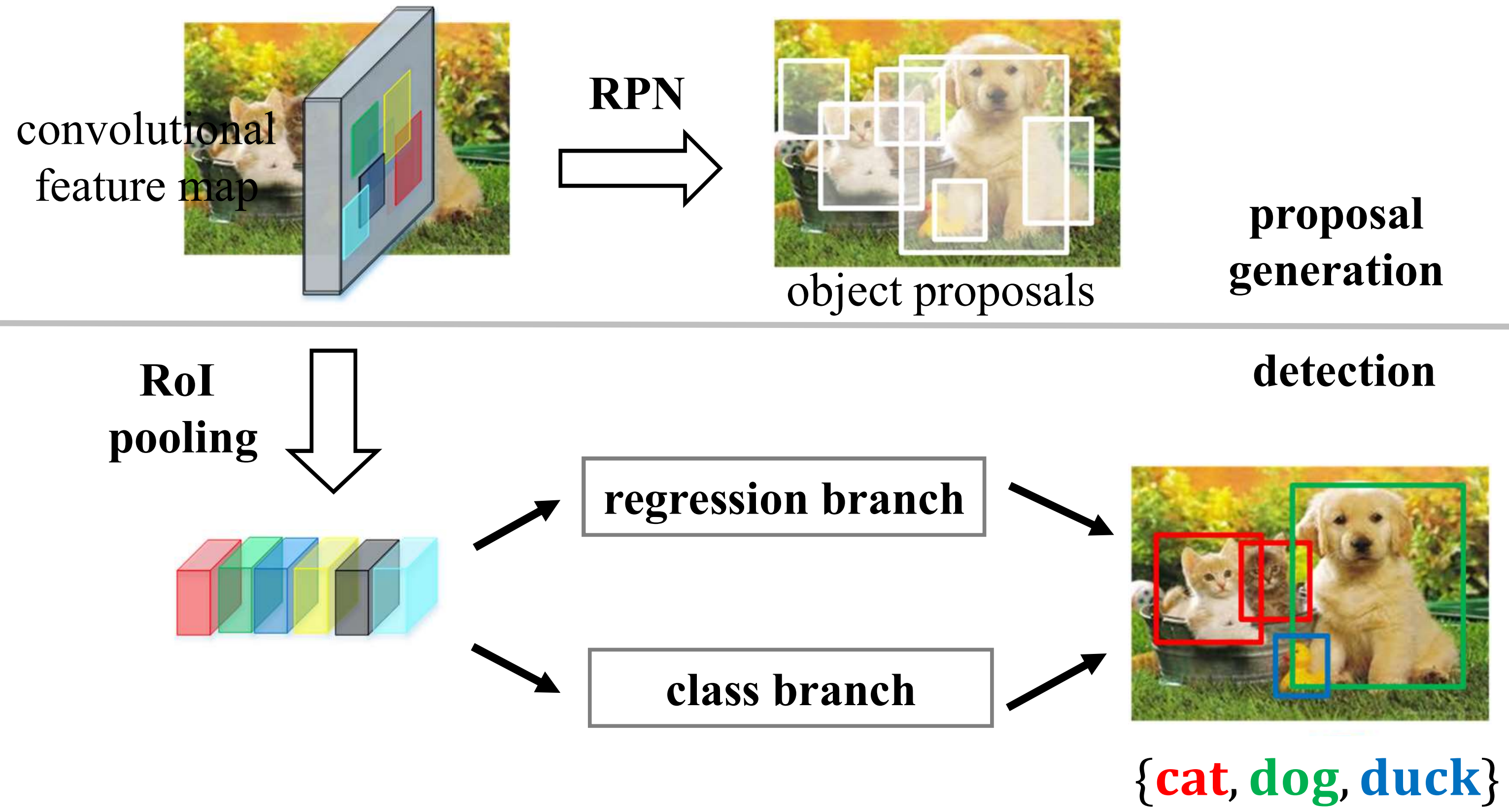
Introduction

Quarter Unit: Faster R-CNN

Link: https://youtu.be/K3C4Jy1X2cA

Web: http://gg.gg/quarter

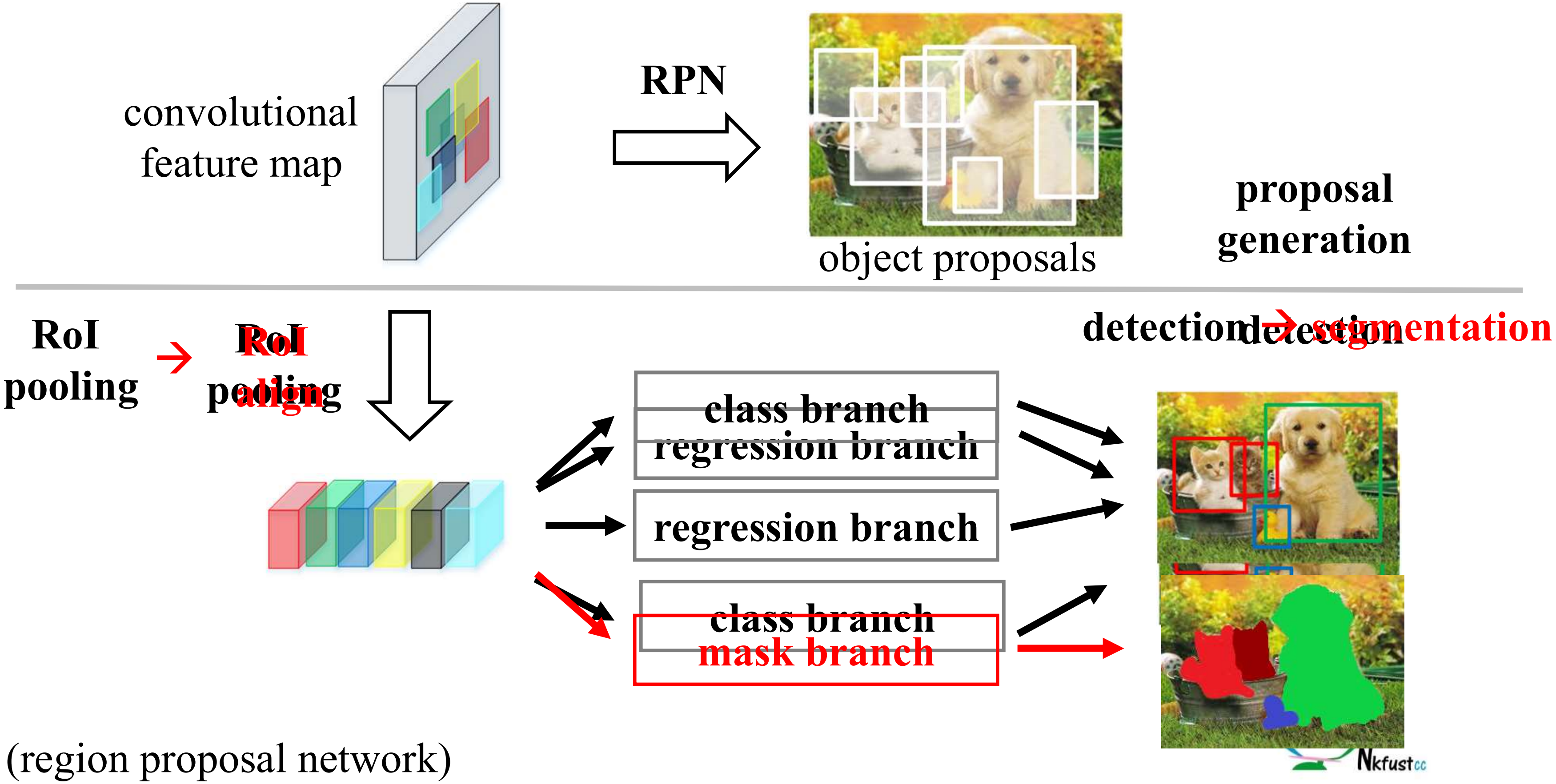
- About Mask R-CNN



RPN (region proposal network)

Introduction

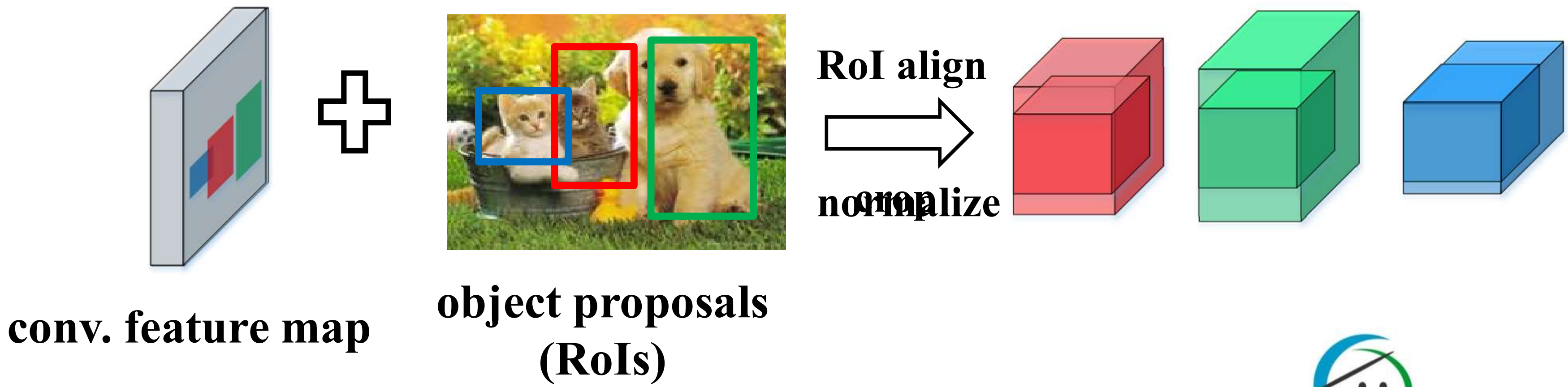
- About Mask R-CNN



RPN (region proposal network)

RoI Align

- Objective
  - extract features within all RoIs for parameter prediction.
    - **<u>crop</u>** the feature maps within the RoIs
    - **<u>normalize</u>** the clipped feature maps



RoI Align

- Objective
  - extract feature maps by **<u>interpolation</u>** instead of quantization in RoI pooling.
    - preserve exact spatial locations
    - improve mask accuracy about 10% to 50%

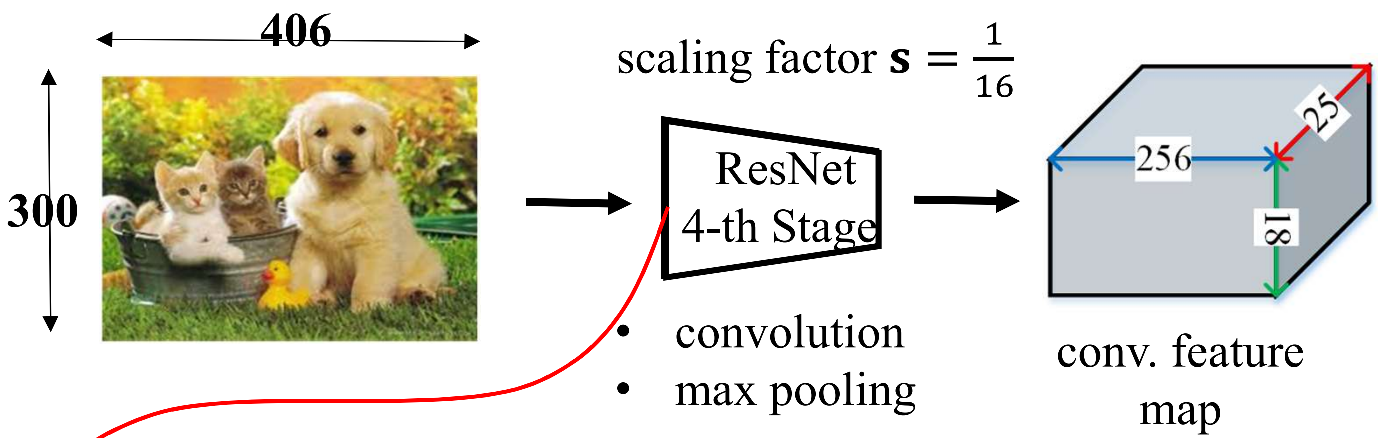
Quarter Unit: RoI Pooling and Align

Link: https://youtu.be/GXYfQsj8RU0

Web: http://gg.gg/quarter

RoI Align

- Formal Statement: Input
  - a feature map from a deep conv. neural network.
  - a list of RoIs from RPN



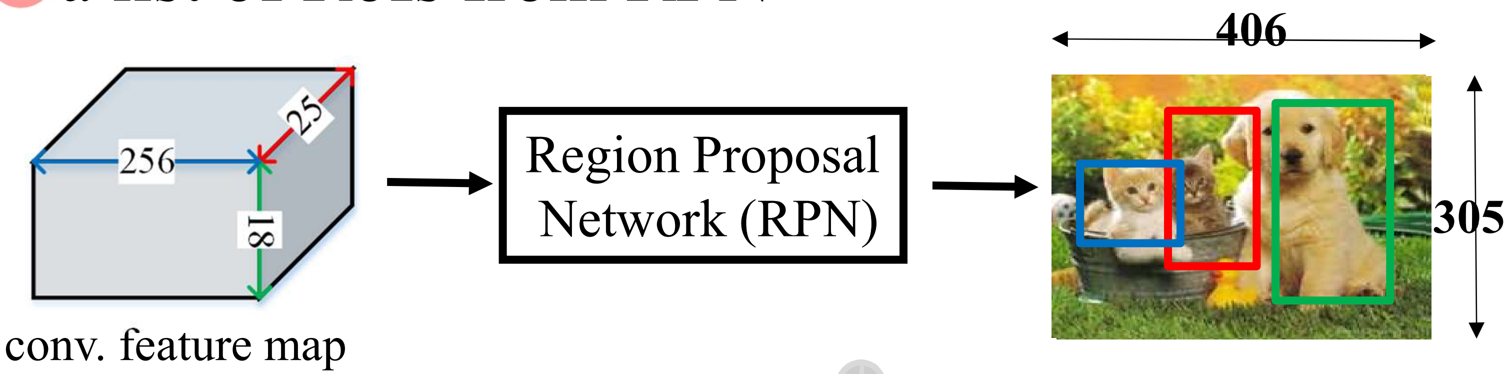
$$\left\lfloor \frac{406}{16} \right\rfloor = 25$$

$$\left\lfloor \frac{300}{16} \right\rfloor = 18$$

Kaiming He, *et. al.*, "Deep Residual Learning for Image Classification", CVPR 2016

RoI Align

- Formal Statement: Input
  - a feature map from a deep conv. neural network.
  - a list of RoIs from RPN



1st RoI: (112, 45), (219, 213)

top-left corner

2nd RoI: (268, 44), (361, 280)

bottom-right corner

3rd RoI: (67, 106), (177, 195)

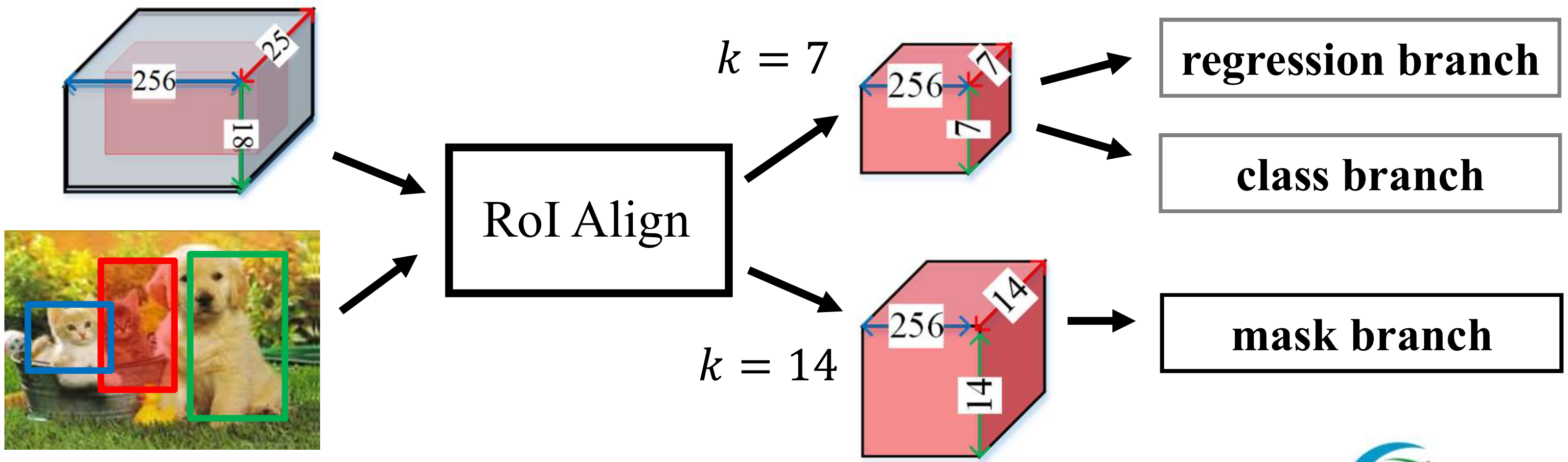
Quarter Unit: RPN

Link: https://youtu.be/0tBhRfEzUWs

Web: http://gg.gg/quarter

RoI Align

- Formal Statement: Output
  - a list of RoI feature maps with $k \times k \times c$
    - k: pre-defined size
    - c: channel number of input conv. feature map

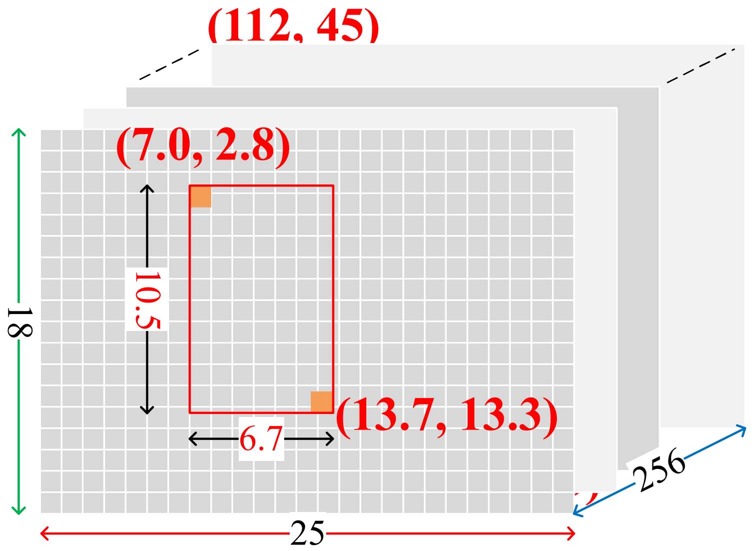


RoI Align

- Align Steps: Overview
  - Step 1 (RoI mapping): multiply the scaling factor to map RoI to conv. feature map
  - Step 2 (RoI division): divide the width/height of mapped RoI by k to have $k \times k$ grids.
  - Step 3 (Interpolation): interpolate the values of all sampling points (each grid has $s \times s$ points)
  - Step 4 (max pooling): find the maximum of all $s \times s$ sampling points in a grid.

RoI Align

- Align Steps
  - RoI mapping: multiple the scaling factor

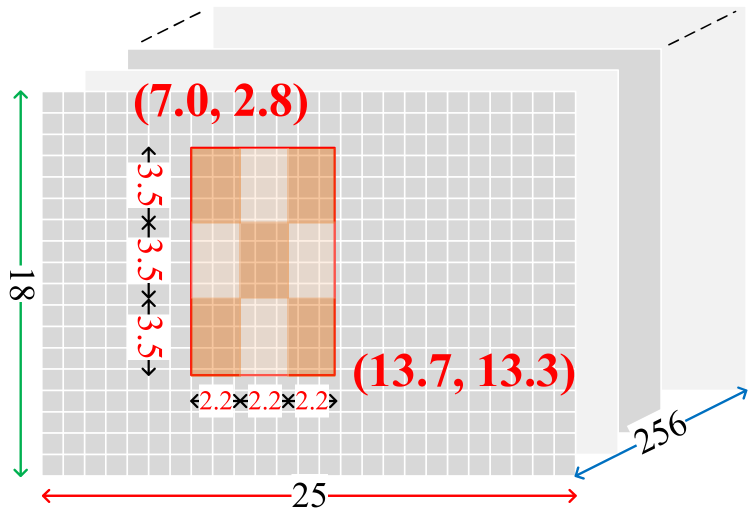


$$\left(112 \times \frac{1}{16}, 45 \times \frac{1}{16}\right) = (7.0, 2.8)$$

$$\left(219 \times \frac{1}{16}, 213 \times \frac{1}{16}\right) = (13.7, 13.3)$$

RoI Align

- Align Steps
  - RoI division: divide width/height by k



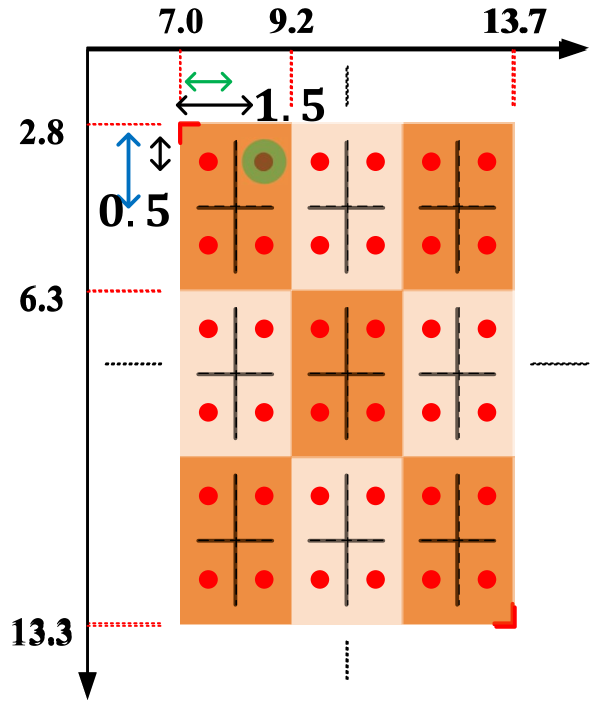
$(\boldsymbol{k} = \mathbf{3})$

grid width: $6.7 \div 3 = \mathbf{2.2}$

grid height: $10.5 \div 3 = \mathbf{3.5}$

RoI Align

- Align Steps: interpolation
  - divide each grid into $s \times s$ cells ($s = 2$)
  - take the centroids of cells as sampling points



$$x: \overbrace{7.0}^{\text{starting coordinate}} + \overbrace{\left(\frac{2.2}{s = 2}\right) \times (1 + 0.5)}^{\text{displacement}} = \mathbf{8.65}$$

$$y: 2.8 + \left(\frac{3.5}{s = 2}\right) \times (0 + 0.5) = \mathbf{3.67}$$

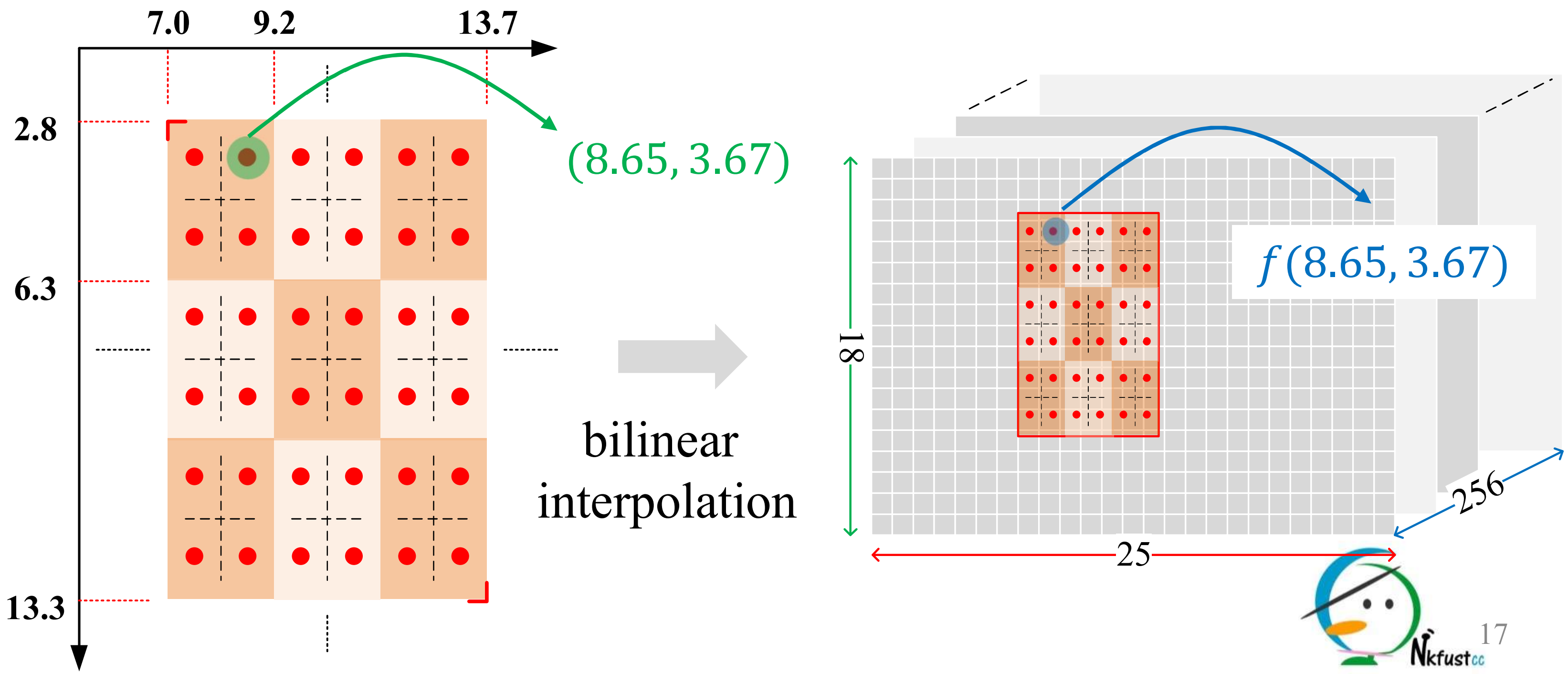
RoI Align

Quarter Unit: RoI Pooling and Align

Link: https://youtu.be/GXYfQsj8RU0

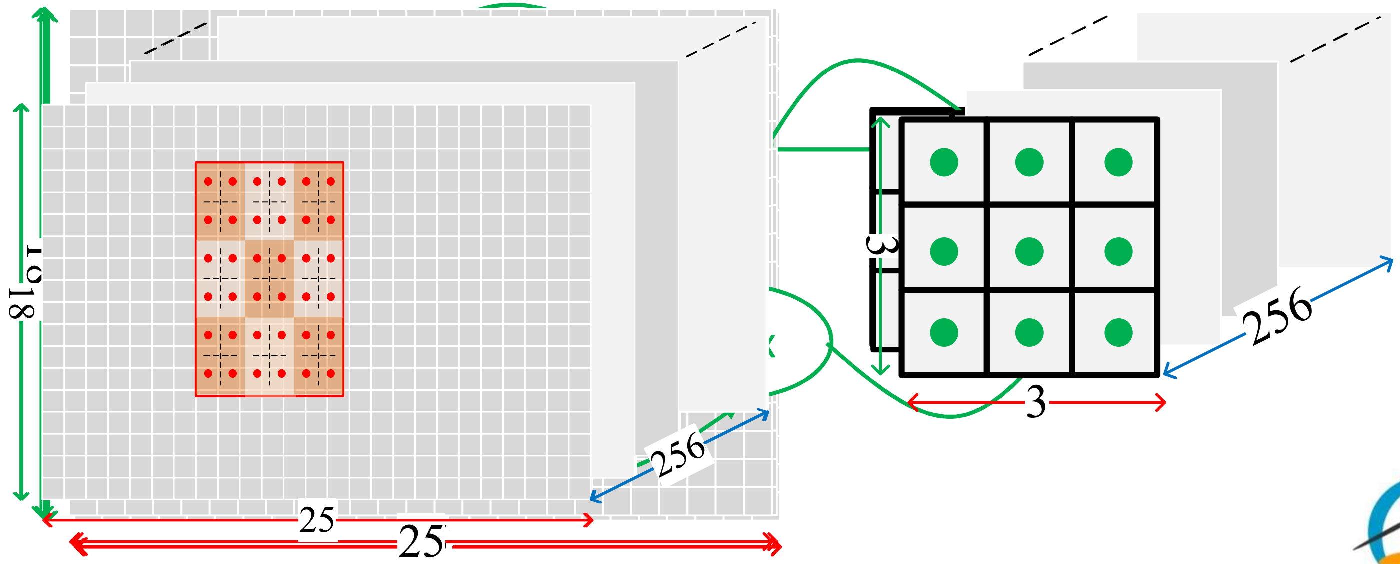
Web: http://gg.gg/quarter

- Align Steps: interpolation
  - interpolate the feature value of every sample point by **bilinear interpolation**.



RoI Align

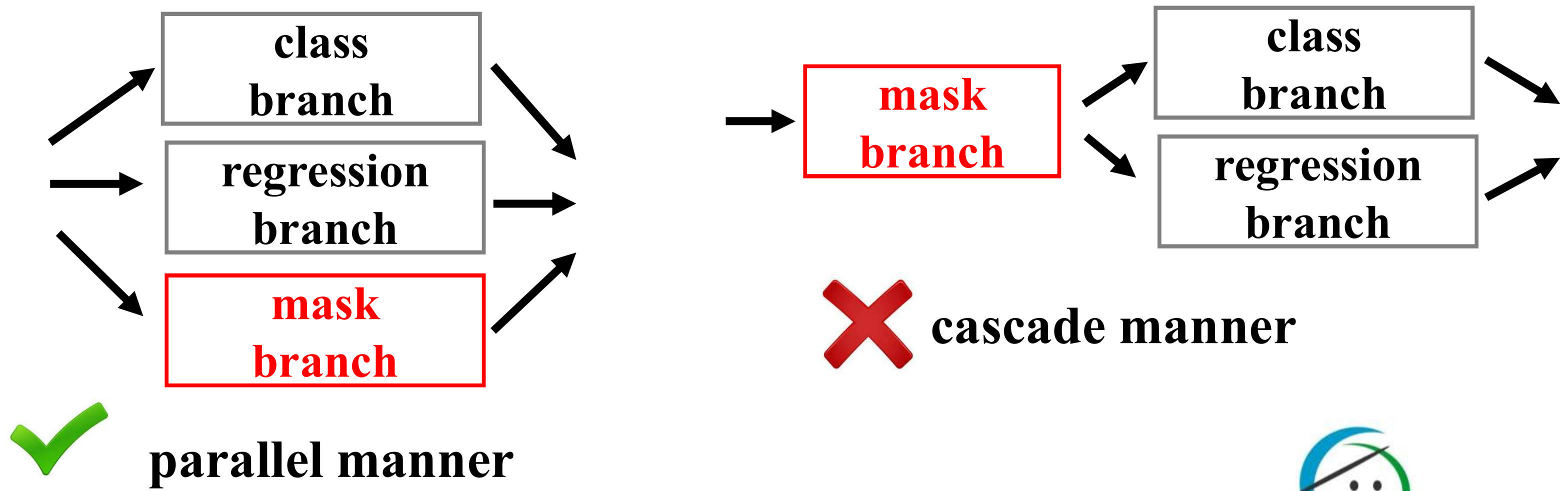
- Align Steps: max pooling
  - take the maximum of feature values of sampling points in each grid
  - aggregate the results in all channels



Mask Branch

- Design Concept
  - follow the spirit of the Fast R-CNN that simplifies the multi-stage pipeline

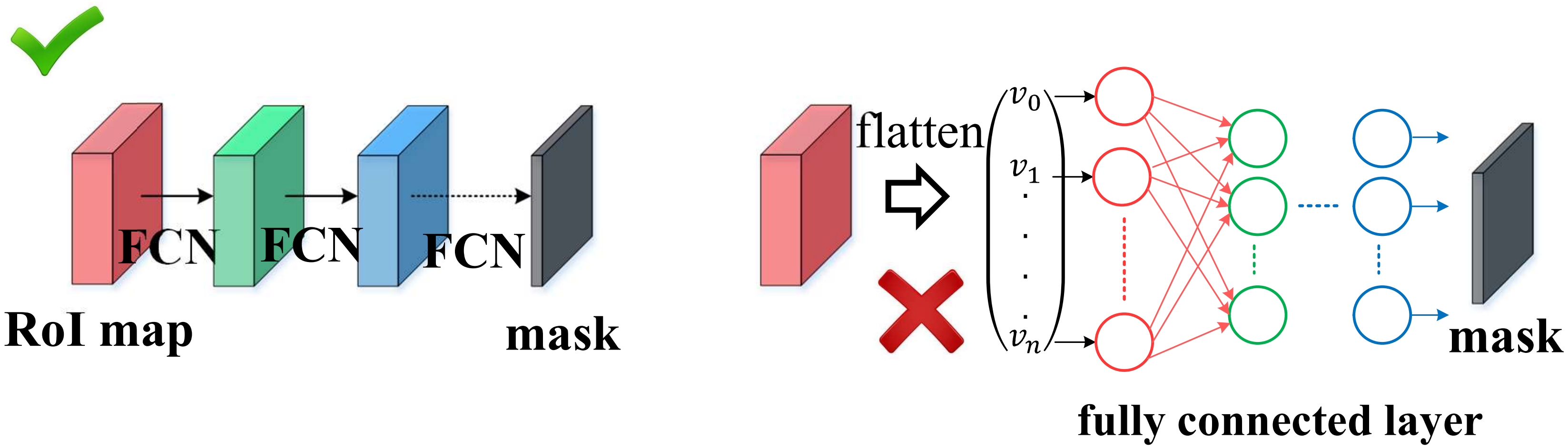
⇨ predict the object mask in **parallel**



Mask Branch

- Design Concept
  - maintain object spatial layout without collapsing it into a vector representation.

⇨ use FCN for mask prediction

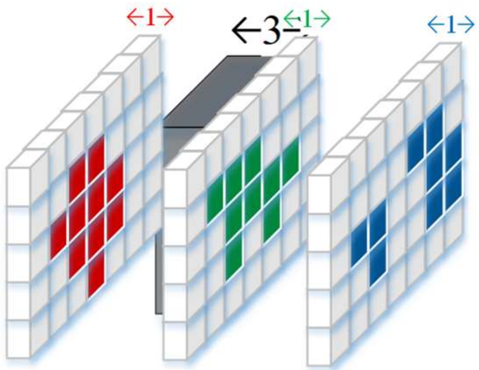


FCN: fully convolutional network

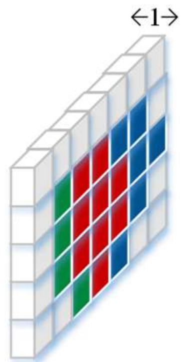
Mask Branch

- Design Concept
  - decouple mask and classification predictions.

⇨ predict a binary mask for each class independently



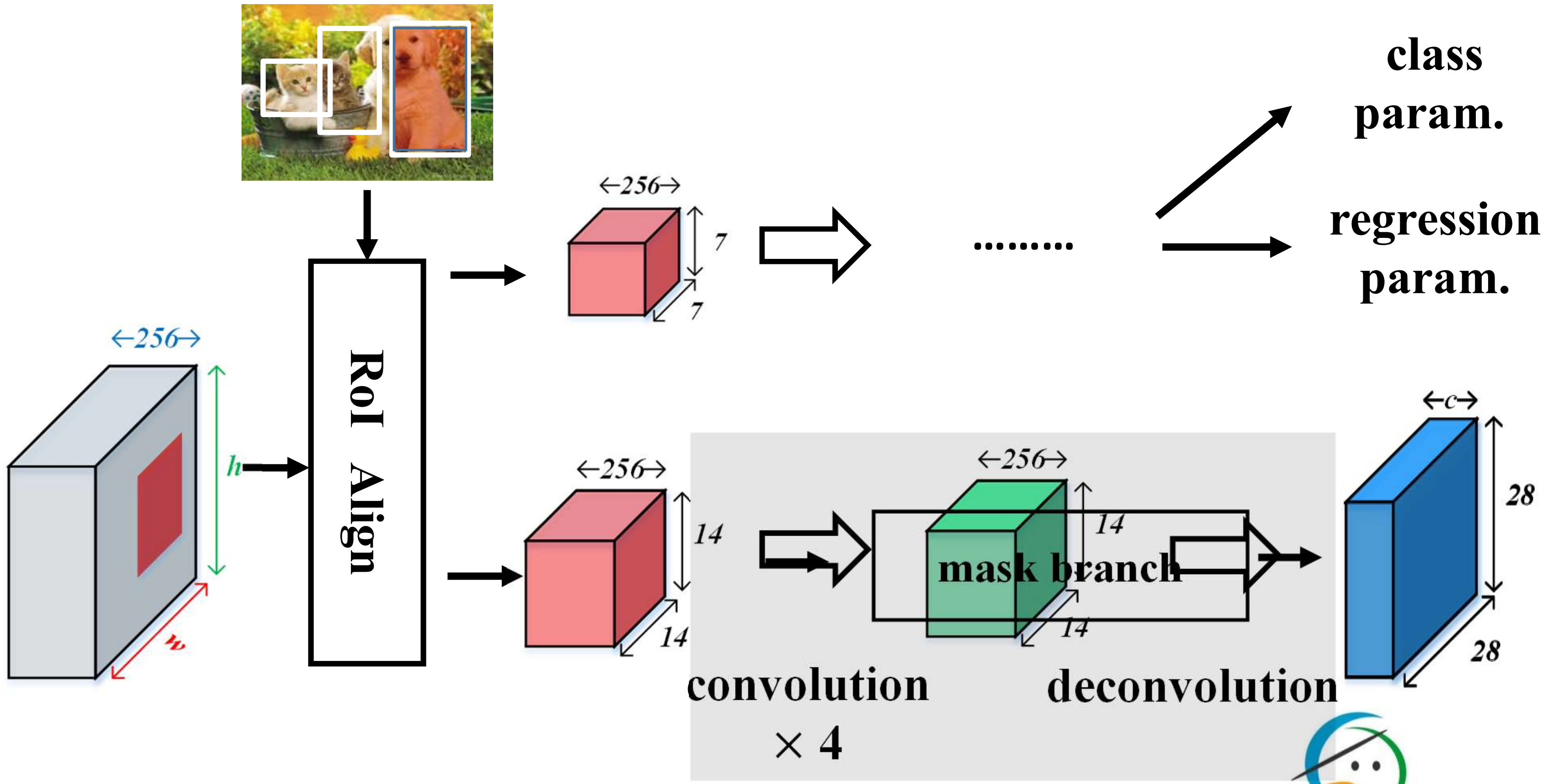
C = {cat, dog, duck}



C = {bg, cat, dog, duck}

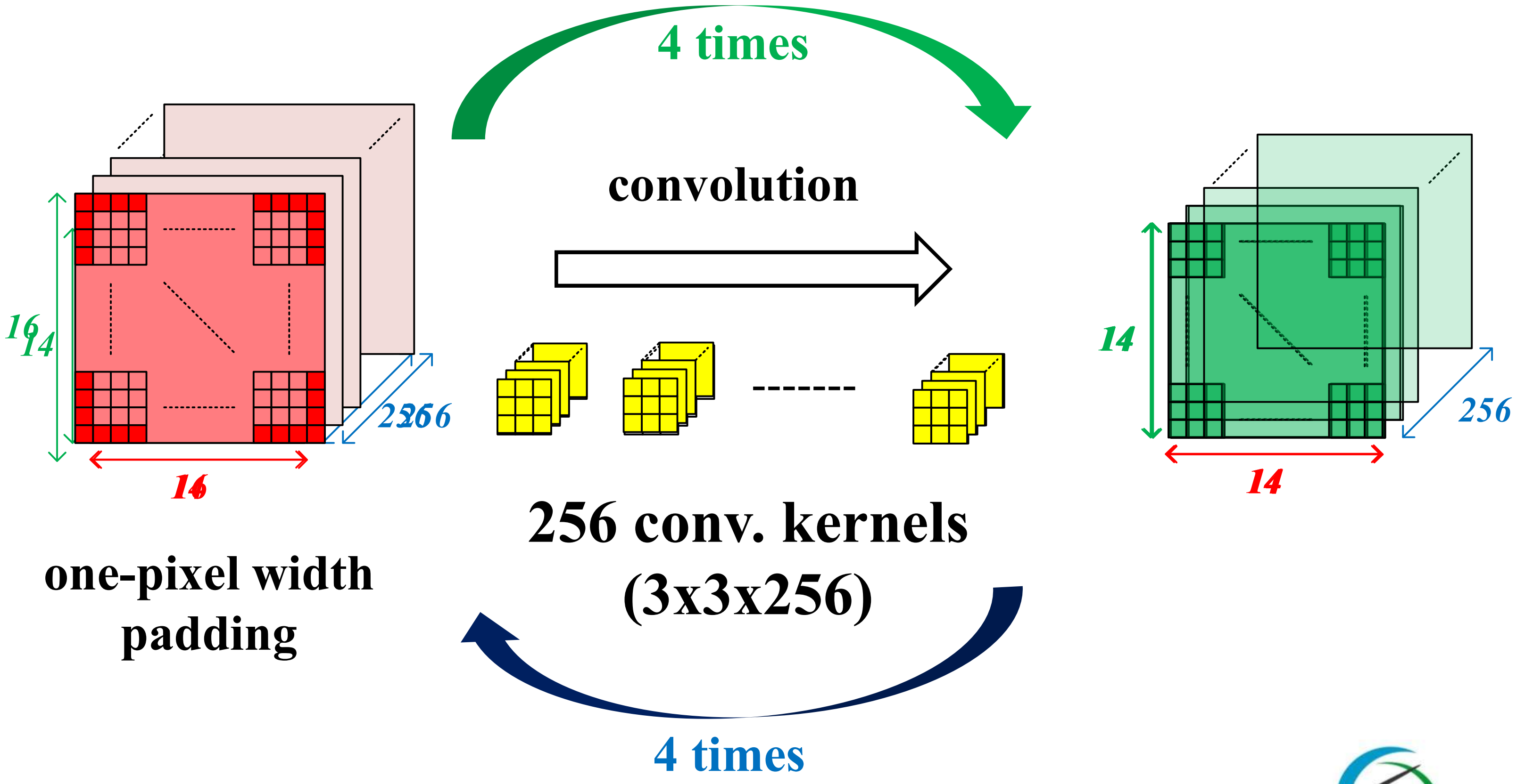
Mask Branch

- Network Architecture



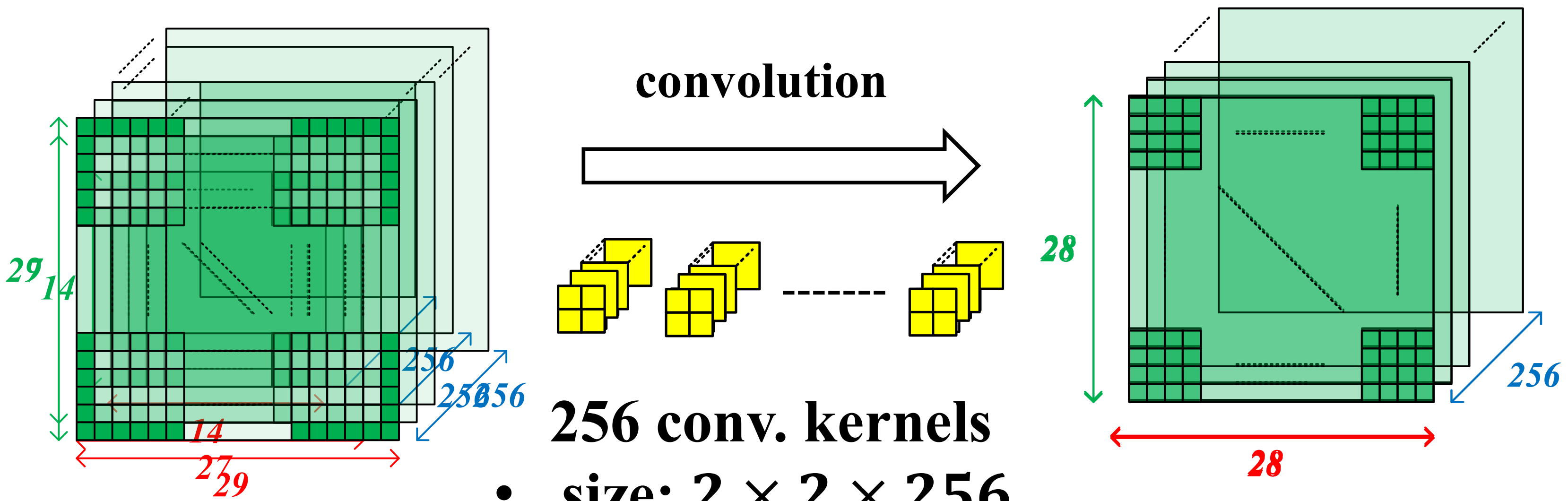
Mask Branch

- Network Architecture: convolution x 4



Mask Branch

- Network Architecture: deconvolution



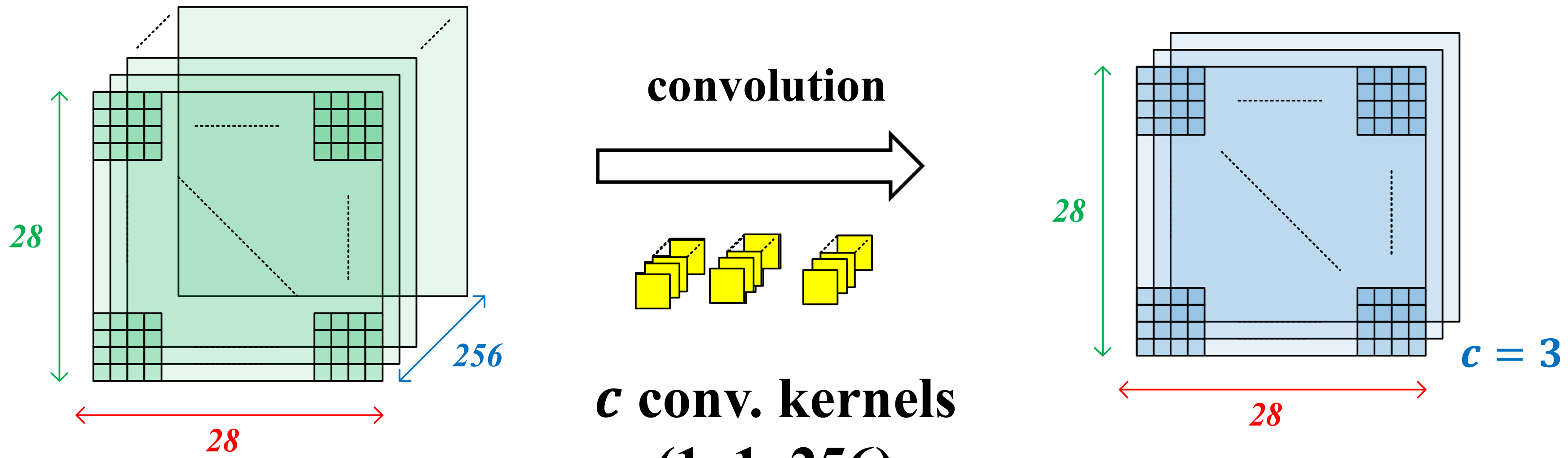
one-pixel width padding

256 conv. kernels

- size: $2 \times 2 \times 256$
- stride: 1×1

Mask Branch

- Network Architecture: deconvolution

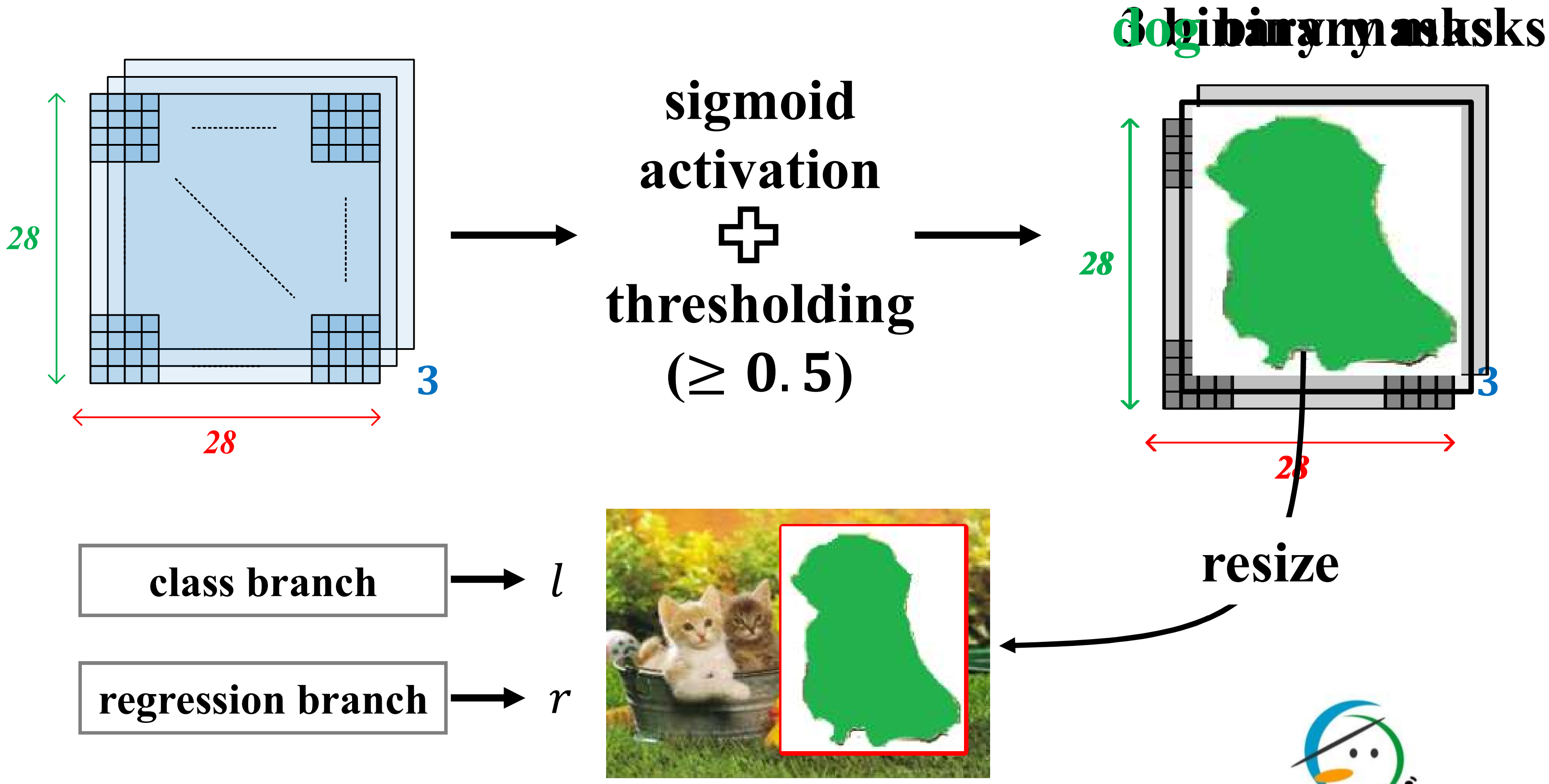


$\boldsymbol{c}$ **conv. kernels**
(1x1x256)

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

Mask Branch

- Network Architecture: mask generation



Mask Branch

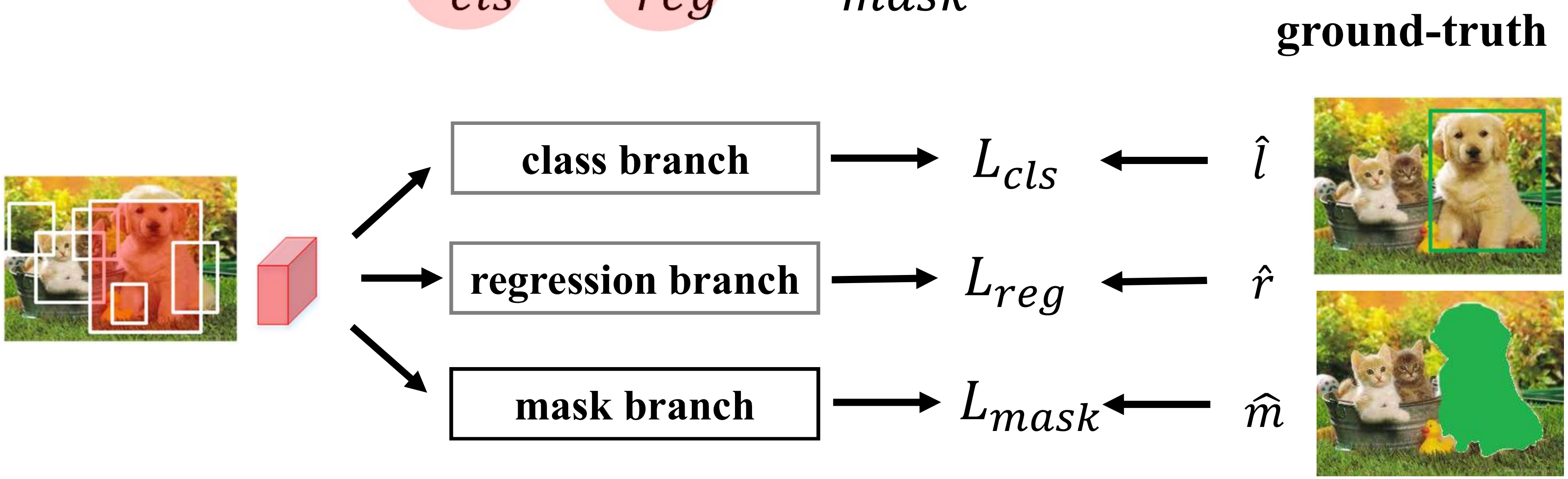
Quarter Unit: Faster R-CNN

Link: https://youtu.be/K3C4Jy1X2cA

Web: http://gg.gg/quarter

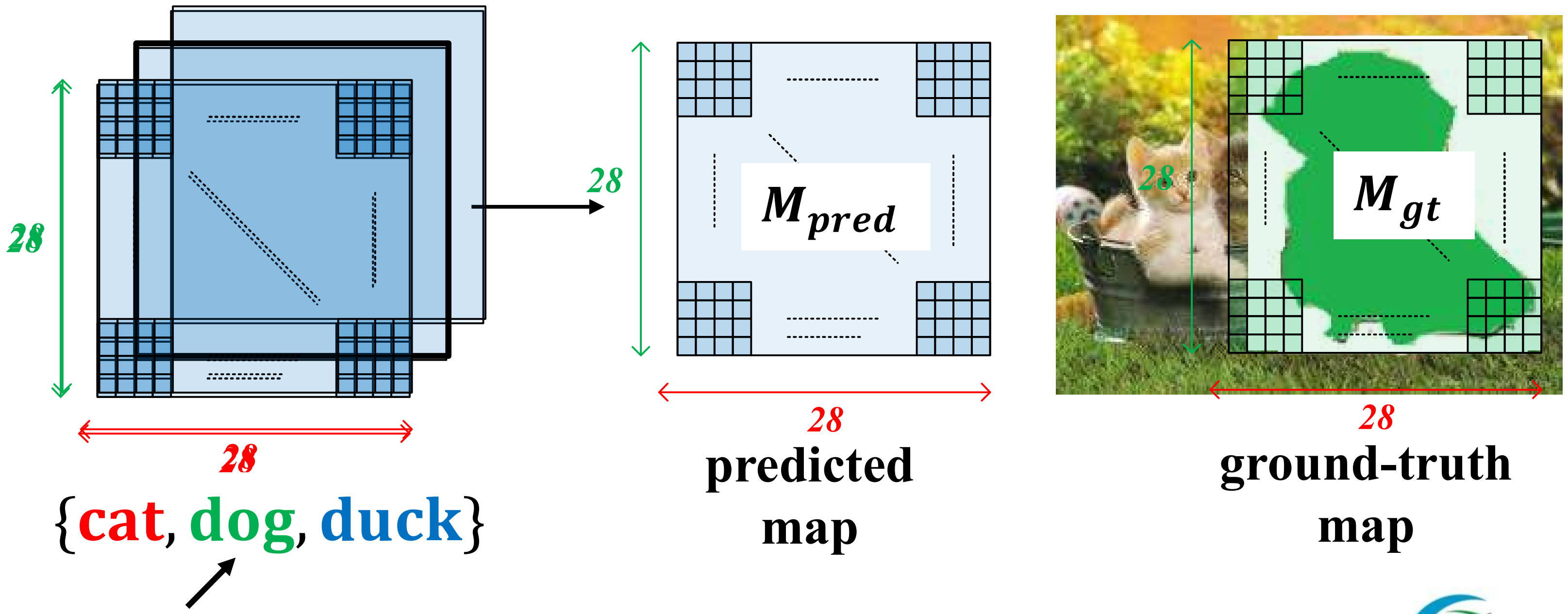
- Mask Loss L_{mask}
  - is part of the total loss L defined on each RoI

$$L = L_{cls} + L_{reg} + L_{mask}$$



Mask Branch

- Mask Loss
  - evaluates the difference between M_{pred} and M_{gt}



Mask Branch

- Mask Loss
  - is defined as average **binary cross-entropy** (BCE) loss.

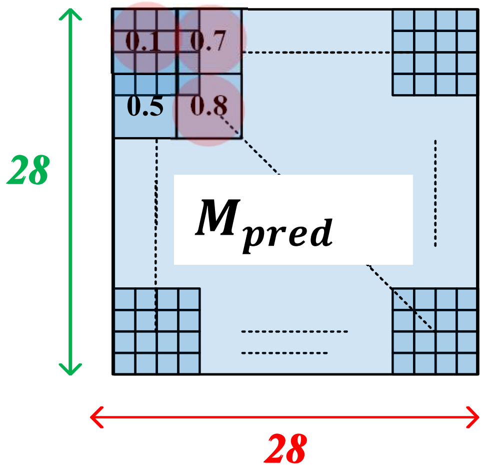
$$L_{mask} = -\frac{1}{N} \times \sum_{(x,y)} \boxed{L_{BCE}(M_{pred}(x,y), M_{gt}(x,y))}$$

no. points in the map

BCE loss of a point (x, y)

$$L_{BCE} = \begin{pmatrix} M_{gt}(x,y) \times \log\left(M_{pred}(x,y)\right) + \\ (1.0 - M_{gt}(x,y)) \times \log\left(1.0 - M_{pred}(x,y)\right) \end{pmatrix}$$

Mask Branch



$$L_{mask} = -\frac{1}{N} \times \sum_{(x,y)} L_{BCE}(M_{pred}(x,y), M_{gt}(x,y))$$

$N = 28 \times 28$

$\Rightarrow 0.0 \times \log(0.1) + (1.0 - 0.0) \times \log(1.0 - 0.1) +$

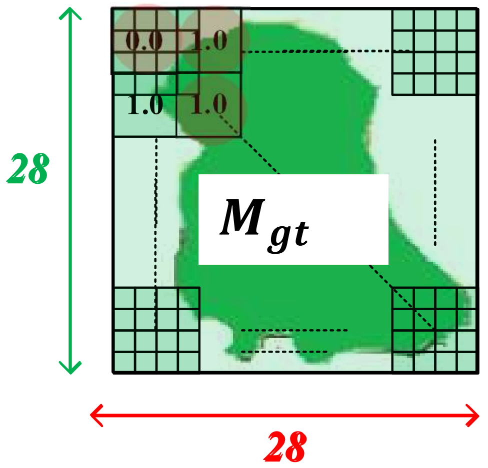
$\Rightarrow 1.0 \times \log(0.7) + (1.0 - 1.0) \times \log(1.0 - 0.7) +$

$\ldots\ldots\ldots\ldots. \quad +$

$1.0 \times \log(0.5) + (1.0 - 1.0) \times \log(1.0 - 0.5) +$

$\Rightarrow 1.0 \times \log(0.8) + (1.0 - 1.0) \times \log(1.0 - 0.8) +$

$\ldots\ldots\ldots\ldots.$




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