Instance Segmentation

Danna Gurari
University of Colorado Boulder
Fall 2024

Review

- Last lecture: object detection
  - Motivation
  - Datasets
  - Evaluation metric
  - Faster R-CNN
  - DETR
  - Discussion
- Assignments (Canvas)
  - Project proposal was due earlier today
  - Reading assignments due next Monday and Wednesday
- Questions?

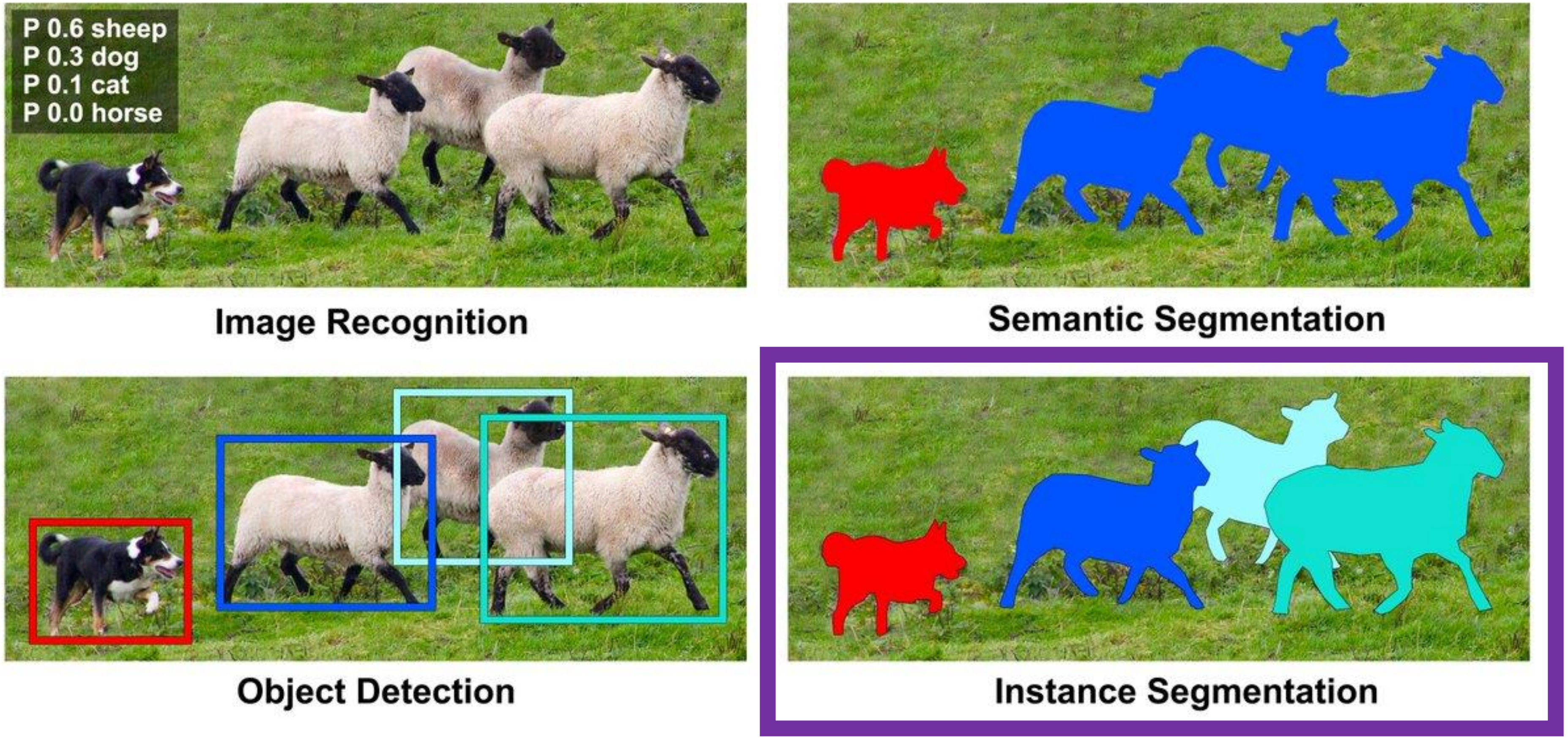
Instance Segmentation: Today’s Topics

- Motivation
- Datasets
- Evaluation metric
- Mask R-CNN
- YOLACT

Instance Segmentation: Today’s Topics

- Motivation
- Datasets
- Evaluation metric
- Mask R-CNN
- YOLACT

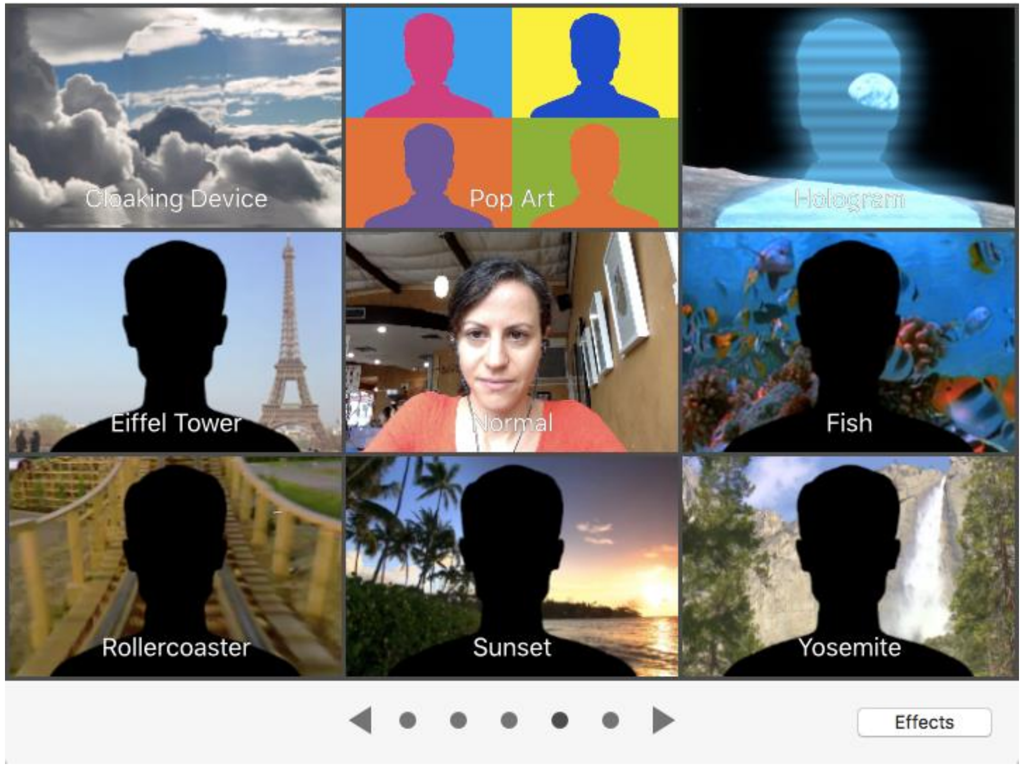
Task: Fuse Semantic Segmentation (and So Classification) with Object Detection



Instances of the same category are separated

https://ai-pool.com/d/could-you-explain-me-how-instance-segmentation-works

Applications (recall those from prior lectures); e.g.,



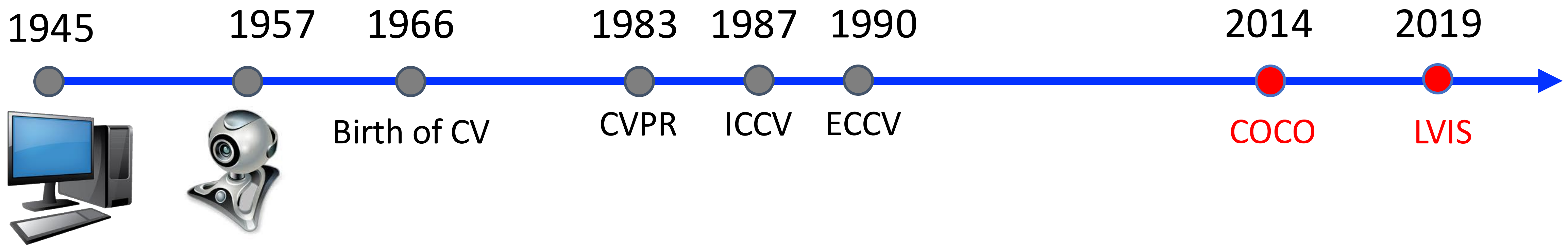
Rotoscoping

Business Traffic Analytics

Instance Segmentation: Today’s Topics

- Motivation
- Datasets
- Evaluation metric
- Mask R-CNN
- YOLACT

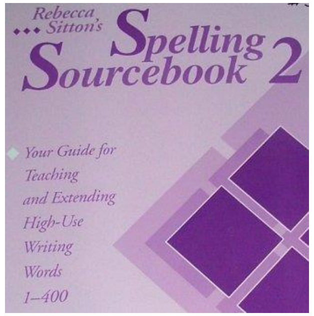
Historical Context



MSCOCO (**C**ommon **O**bjects in **Co**ntext)

1. Category Selection

- 272 candidates from:
1) WordNet, SUN, VOC, …
2) Popular words describing visual objects:



3) 4-8 yr olds listing objects in indoors/outdoors
- 91 categories chosen by author votes + coverage

Include "things": objects that can easily be labeled; e.g., person, chair

Exclude "stuff": objects with no clear boundaries; e.g., sky, grass,

Rationale: primary interest is in precise localization of object instances

Tsung-Yi Lin et al. Microsoft COCO: Common Objects in Context. ECCV 2014

MSCOCO

Selected 91 from 272 categories in bold (without *)

person	**bicycle**	**car**	**motorcycle**	**bird**	**cat**	**dog**	**horse**	**sheep**	**bottle**
chair	**couch**	**potted plant**	**tv**	**cow**	**airplane**	**hat***	license plate	**bed**	**laptop**
fridge	**microwave**	**sink**	**oven**	**toaster**	**bus**	**train**	**mirror***	**dining table**	**elephant**
banana	bread	**toilet**	**book**	**boat**	**plate***	**cell phone**	**mouse**	**remote**	**clock**
face	hand	**apple**	**keyboard**	**backpack**	steering wheel	**wine glass**	chicken	**zebra**	**shoe***
eye	mouth	**scissors**	**truck**	**traffic light**	**eyeglasses***	**cup**	**blender***	**hair drier**	wheel
street sign*	**umbrella**	**door***	**fire hydrant**	**bowl**	teapot	**fork**	**knife**	**spoon**	**bear**
headlights	**window***	**desk***	computer	**refrigerator**	**pizza**	squirrel	duck	**frisbee**	guitar
nose	**teddy bear**	**tie**	**stop sign**	**surfboard**	**sandwich**	pen/pencil	**kite**	**orange**	**toothbrush**
printer	pans	head	**sports ball**	**broccoli**	**suitcase**	**carrot**	chandelier	**parking meter**	fish
handbag	**hot dog**	stapler	basketball hoop	**donut**	**vase**	**baseball bat**	**baseball glove**	**giraffe**	jacket
skis	**snowboard**	table lamp	egg	door handle	power outlet	hair	tiger	table	coffee table
skateboard	helicopter	tomato	tree	bunny	pillow	**tennis racket**	**cake**	feet	**bench**
chopping board	washer	lion	monkey	**hair brush***	light switch	arms	legs	house	cheese
goat	magazine	key	picture frame	cupcake	fan (ceil/floor)	frogs	rabbit	owl	scarf
ears	home phone	pig	strawberries	pumpkin	van	kangaroo	rhinoceros	sailboat	deer
playing cards	towel	hyppo	can	dollar bill	doll	soup	meat	window	muffins
tire	necklace	tablet	corn	ladder	pineapple	candle	desktop	carpet	cookie
toy cars	bracelet	bat	balloon	gloves	milk	pants	wheelchair	building	bacon
box	platypus	pancake	cabinet	whale	dryer	torso	lizard	shirt	shorts
pasta	grapes	shark	swan	fingers	towel	side table	gate	beans	flip flops
moon	road/street	fountain	fax machine	bat	hot air balloon	cereal	seahorse	rocket	cabinets
basketball	telephone	movie (disc)	football	goose	long sleeve shirt	short sleeve shirt	raft	rooster	copier
radio	fences	goal net	toys	engine	soccer ball	field goal posts	socks	tennis net	seats
elbows	aardvark	dinosaur	unicycle	honey	legos	fly	roof	baseball	mat
ipad	iphone	hoop	hen	back	table cloth	soccer nets	turkey	pajamas	underpants
goldfish	robot	crusher	animal crackers	basketball court	horn	firefly	armpits	nectar	super hero costume
jetpack	robots								

Tsung-Yi Lin et al. Microsoft COCO: Common Objects in Context. ECCV 2014

MSCOCO

1. Category Selection

- 272 candidate categories chosen from:
  1) WordNet, SUN, VOC, …
  2) Most frequent words describing visual objects
  3) 4-8 yr olds listing objects in indoors/outdoors

- 91 categories chosen by author votes + coverage

2. Image Collection

- Images scraped from Flickr because it is believed to often have non-iconic images

- Query: object + object or scene + scene

- Query: unusual categories

- Crowd workers flagged images with multiple objects

Iconic images commonly retrieved with Google, Bing, etc:

(a) Iconic object images (b) Iconic scene images

<u>Goal</u>: images with **contextual** information and taken from **non-canonical** viewpoints

(c) Non-iconic images

Tsung-Yi Lin et al. Microsoft COCO: Common Objects in Context. ECCV 2014

MSCOCO: 2 Tasks

Grids of 128 images:

Tsung-Yi Lin et al. Microsoft COCO: Common Objects in Context. ECCV 2014

MSCOCO Summary

1. Category Selection

- 272 candidates from:
1) WordNet, SUN, VOC, …
2) Popular words describing visual objects
3) 4-8 yr olds listing objects in indoors/outdoors

- 91 categories chosen by author votes + coverage

→

2. Image Collection

- Images scraped from Flickr because it is believed to often have non-iconic images

- Query: object + object or scene + scene

- Query: unusual categories

- Crowd workers flagged images with multiple objects

→

3. Image Annotation

Crowdworkers demarcated specific object types

~1.2M instance segmentations across 188k training, 5k validation, and 41k test images

Tsung-Yi Lin et al. Microsoft COCO: Common Objects in Context. ECCV 2014

Task Decomposition

1. Category Assignment

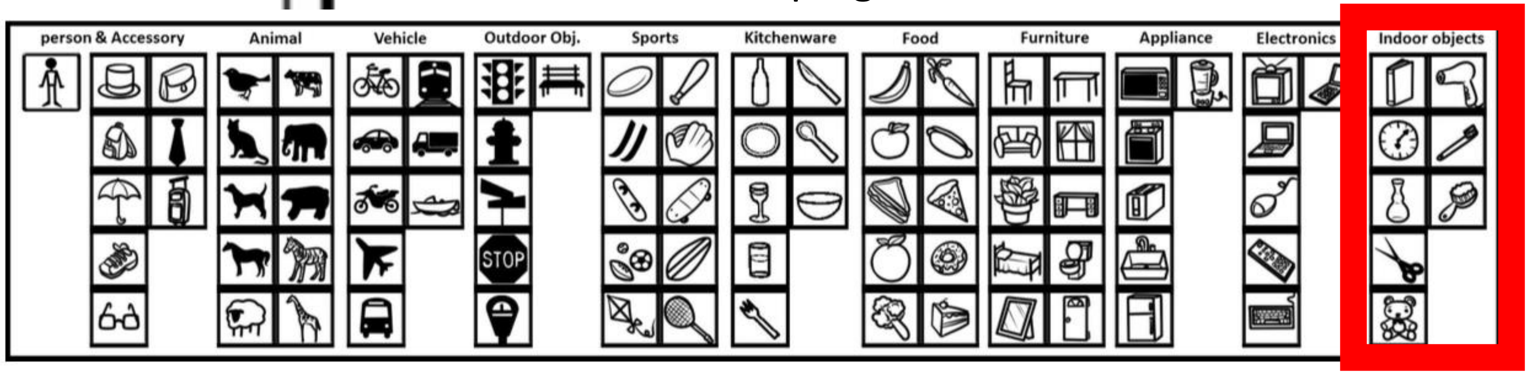
- Crowdworkers identified categories in each image by locating one instance of each

Tsung-Yi Lin et al. Microsoft COCO: Common Objects in Context. ECCV 2014

Category Assignment Task

For high recall, 8 people did this task for each image

11 Groupings



Task Decomposition

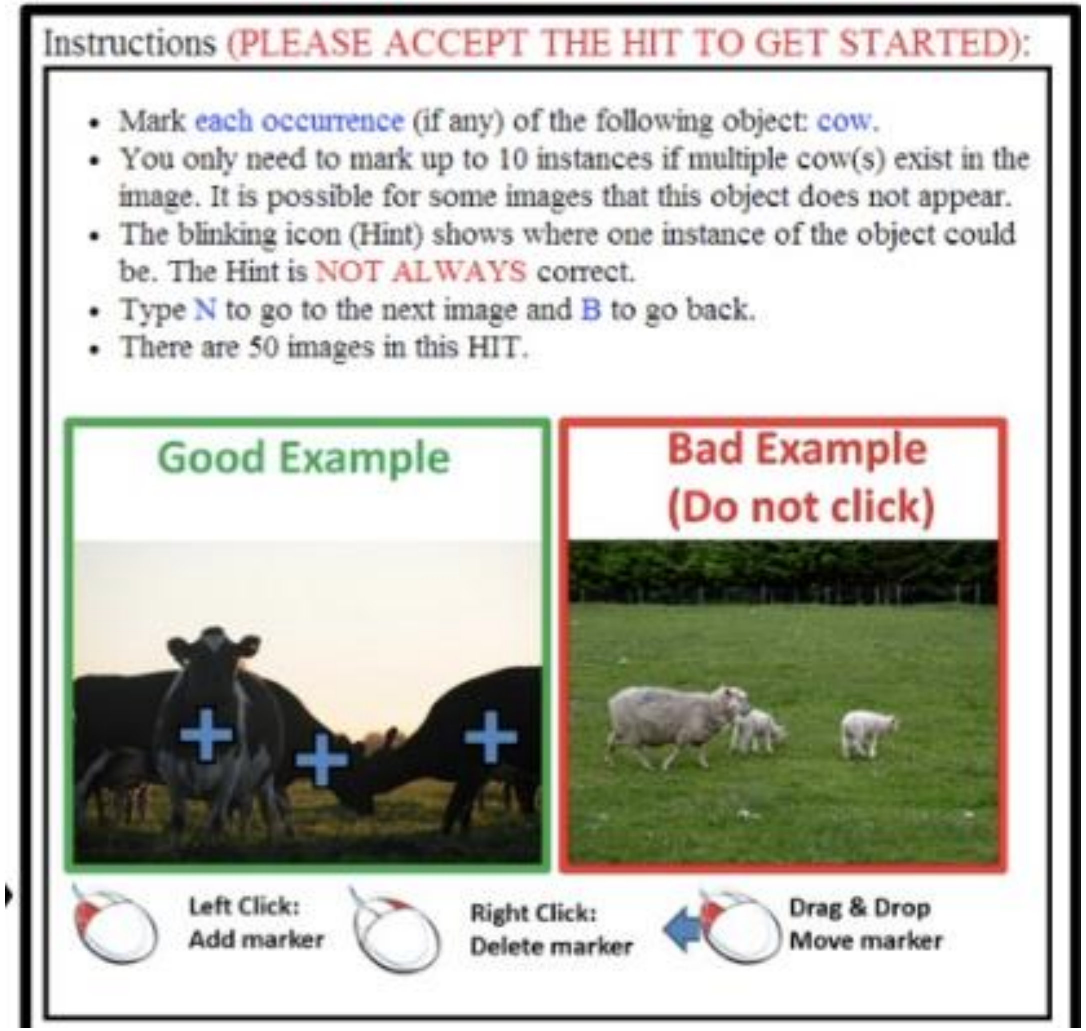
1. Category Assignment

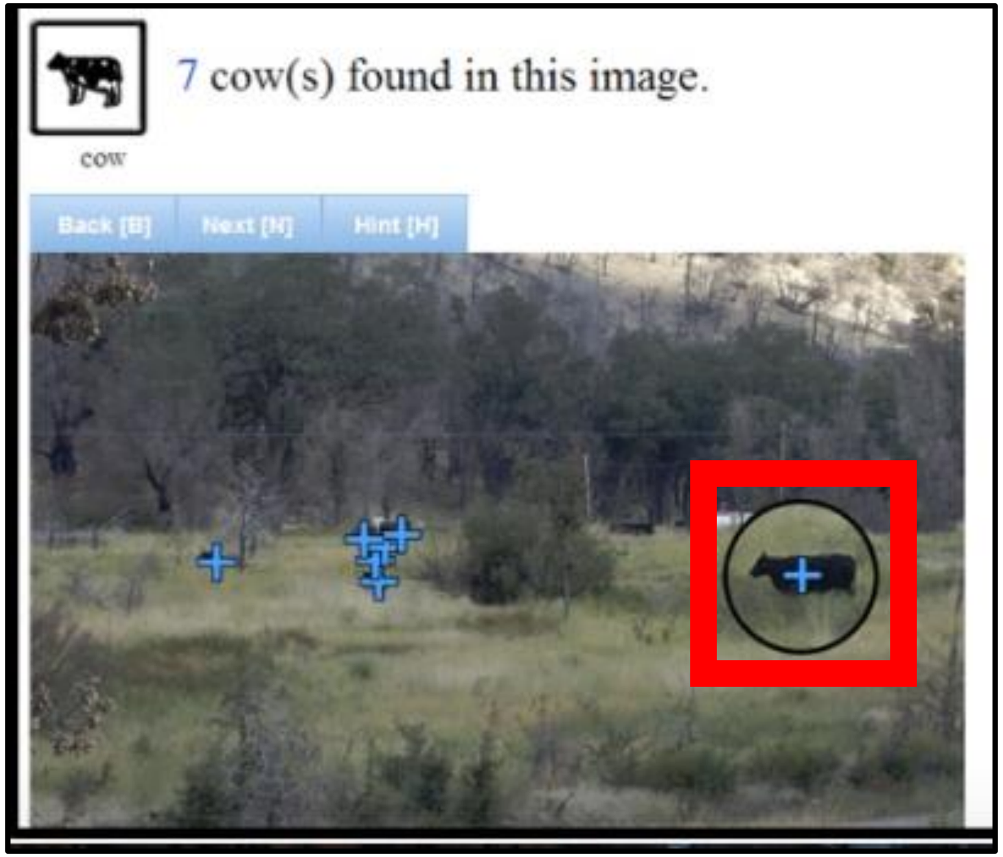
- Crowdworkers identified categories in each image by locating one instance of each

2. Instance Tagging

- Crowdworkers located each instance of the "thing"

Instance Tagging Task





"magnifying glass" feature: doubles resolution to assist with small objects

Tsung-Yi Lin et al. Microsoft COCO: Common Objects in Context. ECCV 2014

Task Decomposition

1. Category Assignment

- Crowdworkers identified categories in each image by locating one instance of each

→

2. Instance Tagging

- Crowdworkers located each instance of the "thing"

→

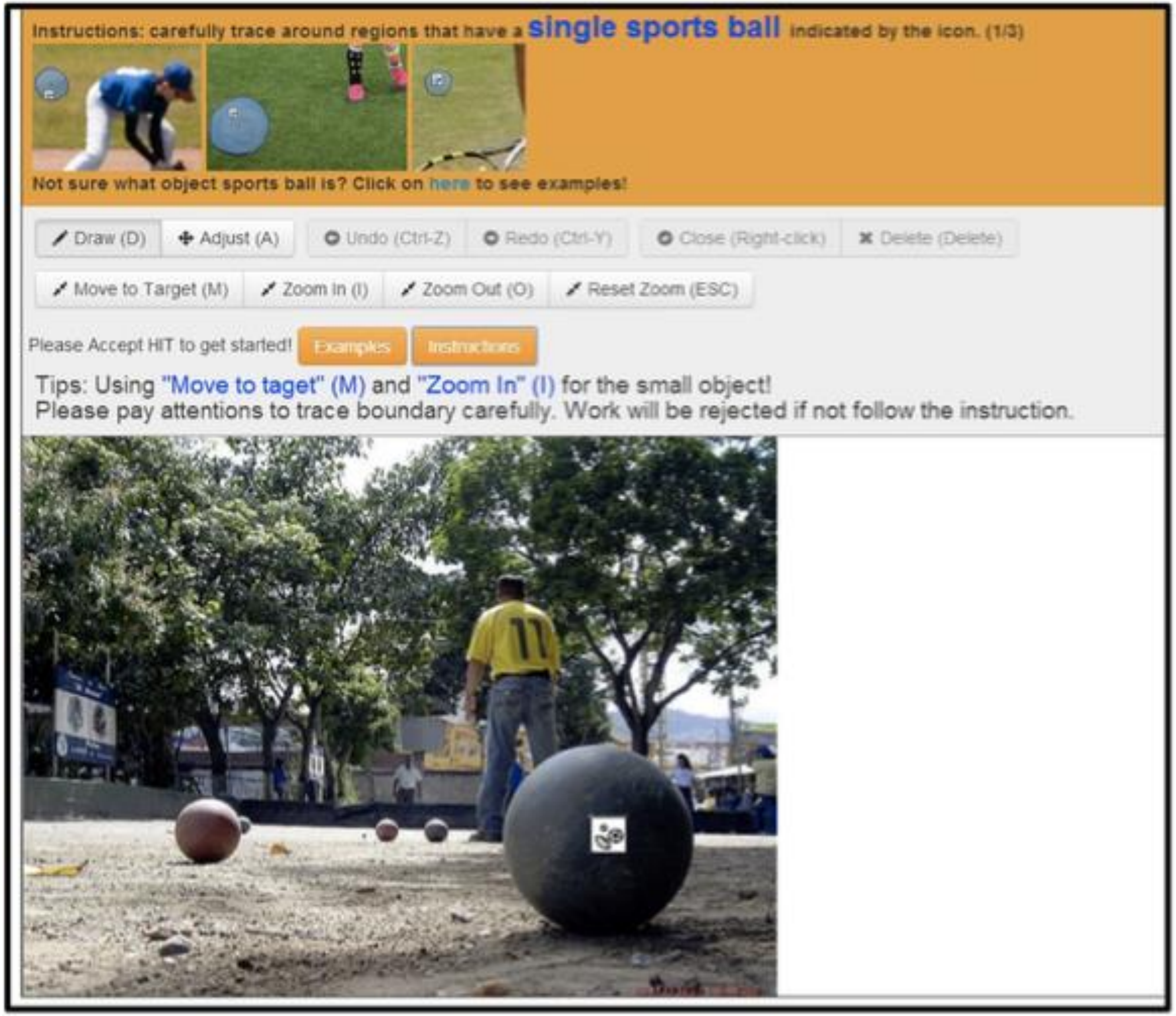
3. Object Segmentation

- Crowdworkers demarcate specified object(s)

- Other crowdworkers verify quality of segmentations

Tsung-Yi Lin et al. Microsoft COCO: Common Objects in Context. ECCV 2014

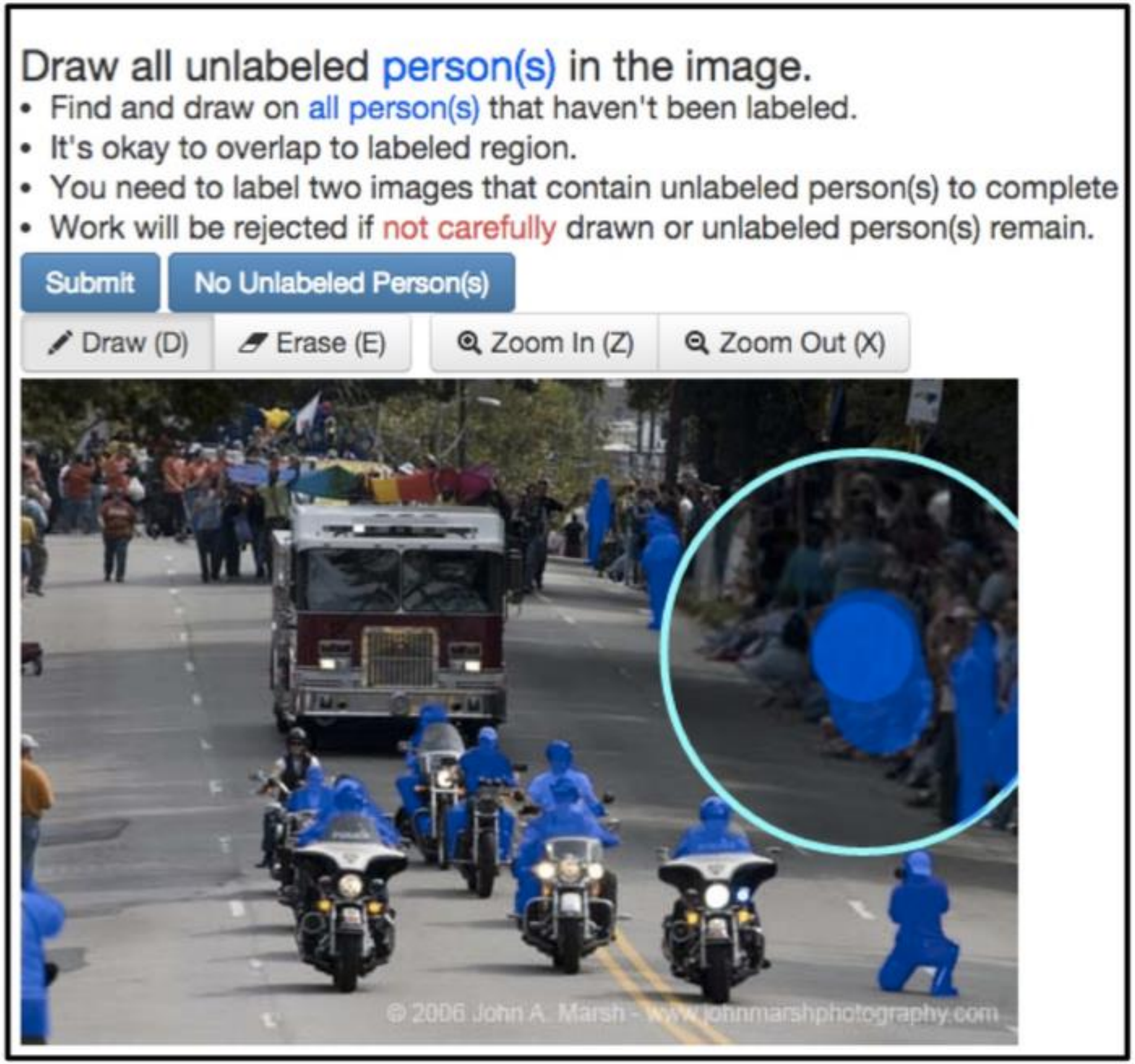
Object Seg.



(Training task per object category required)

Object Seg.

Crowd annotations are done as semantic segmentations (no instances) for images with 10+ instances of an object category.



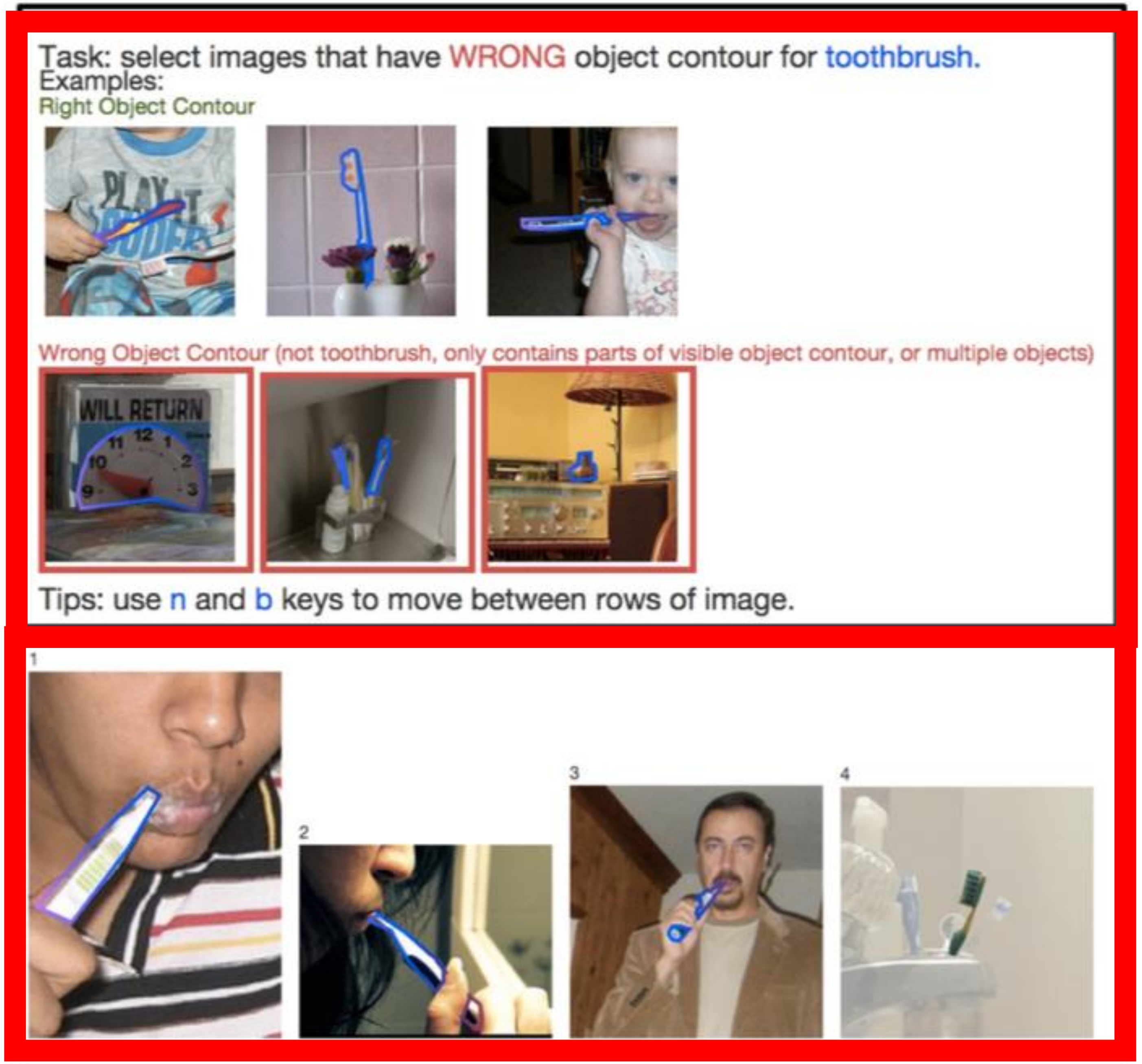
Quality Control

Seeded gold standards: 4 of 64 segmentation known to be bad; a worker had to identify 3 of the 4 known bad segmentations to complete the task.

Verification step: 3-5 workers judged each segmentation's quality.

Blocked workers: regular poor segmentations led to workers being blocked and their work not used.

64 examples



MSCOCO Summary

1. Category Selection

- 272 candidates from:
1) WordNet, SUN, VOC, …
2) Popular words describing visual objects
3) 4-8 yr olds listing objects in indoors/outdoors

- 91 categories chosen by author votes + coverage

2. Image Collection

- Images scraped from Flickr because it is believed to often have non-iconic images

- Query: object + object or scene + scene

- Query: unusual categories

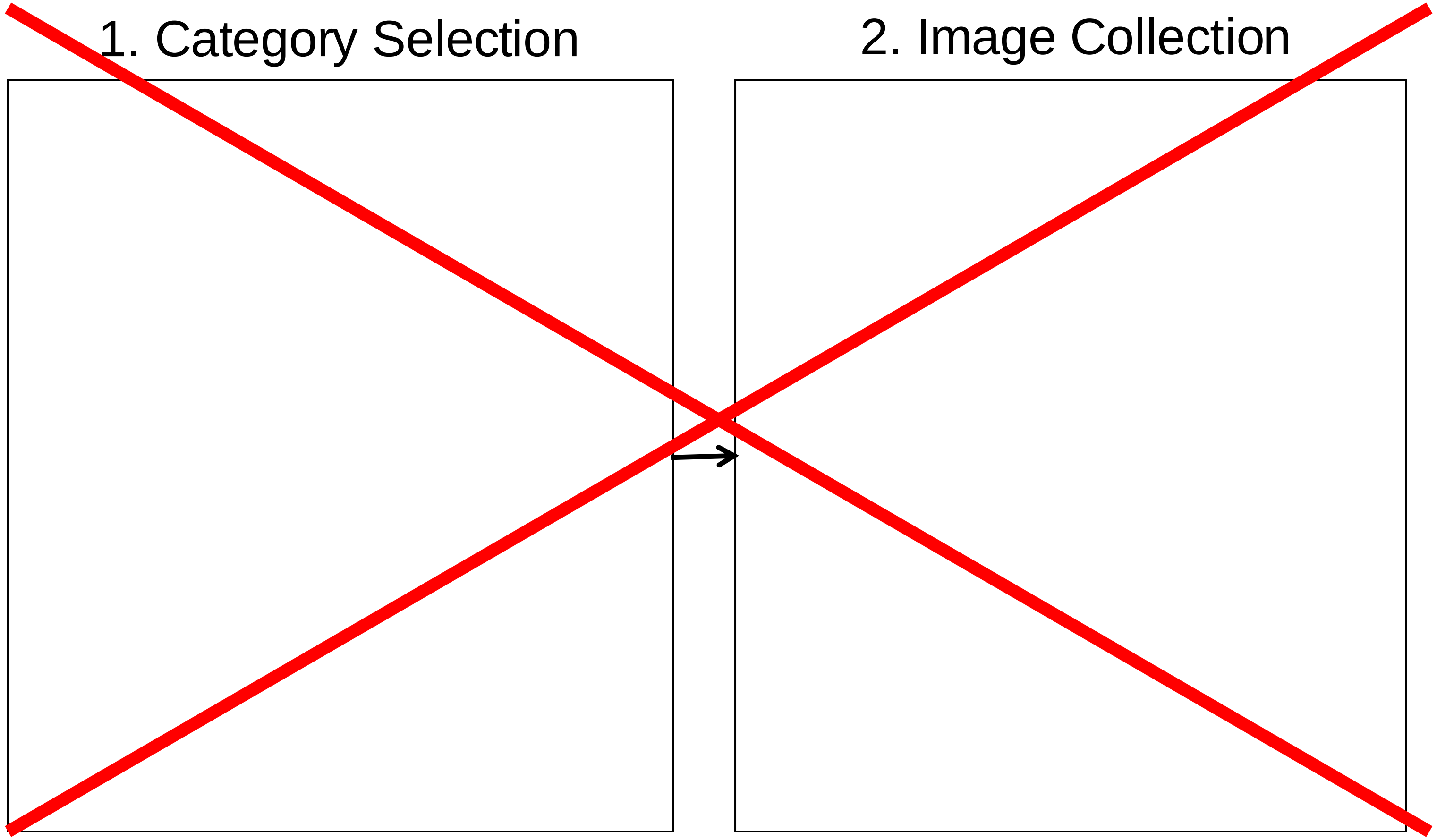
- Crowd workers flagged images with multiple objects

3. Image Annotation

Crowdworkers demarcated specific object types

Tsung-Yi Lin et al. Microsoft COCO: Common Objects in Context. ECCV 2014

LVIS (**L**arge **V**ocabulary **I**nstance **S**egmentation)



Key difference: uses images without pre-specifying categories to annotate

Resulted in ~2M instance segmentations spanning 1203 categories (some rare) for ~160k COCO images

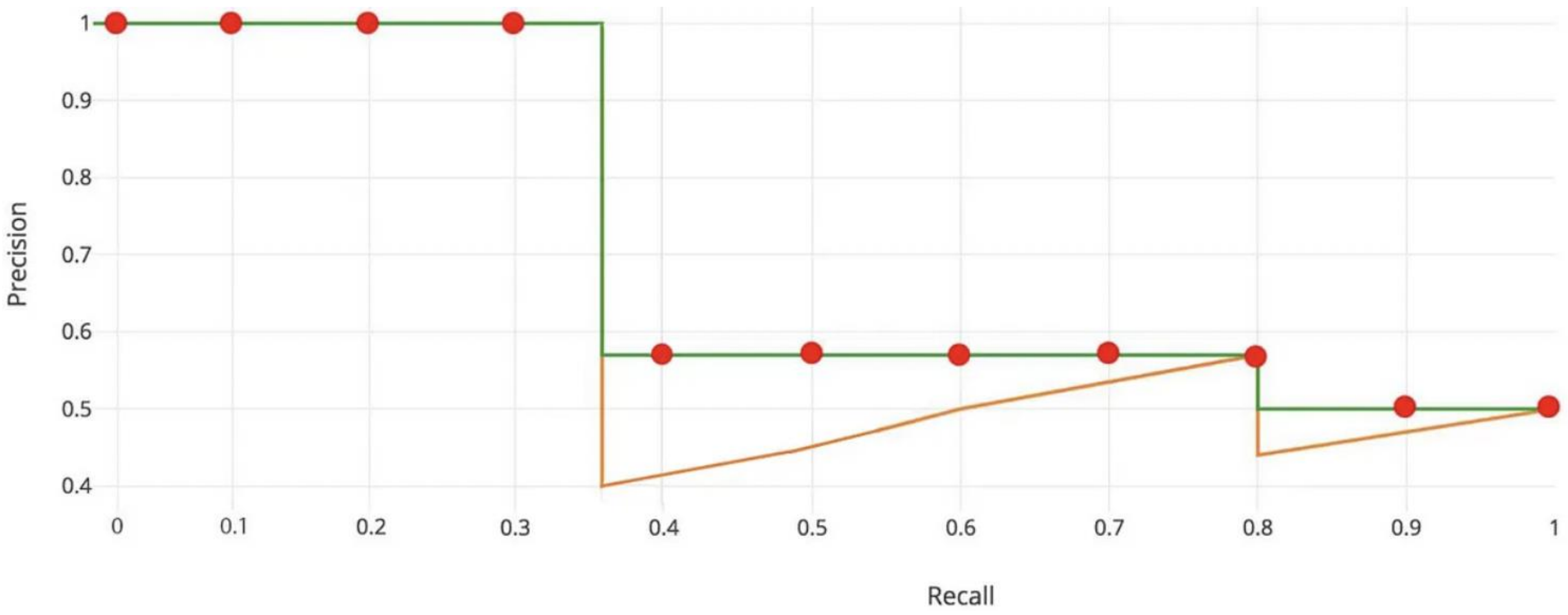
Gupta, Dollar, and Girshick. LVIS: A Dataset for Large Vocabulary Instance Segmentation. CVPR 2019

Instance Segmentation: Today’s Topics

- Motivation
- Datasets
- Evaluation metric
- Mask R-CNN
- YOLACT

Recall: Mean Average Precision (mAP)

- **Mean per-category average precision**: area under precision-recall curve for a category created by varying confidence level determining a positive prediction (using maximum precision value to the right)



We plot precision-recall points using all confidence values predicted by a model for a category.

We then interpolate between the points and compute the area under the curve.

https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173

Great tutorial: Padilla et al. A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit. 2021

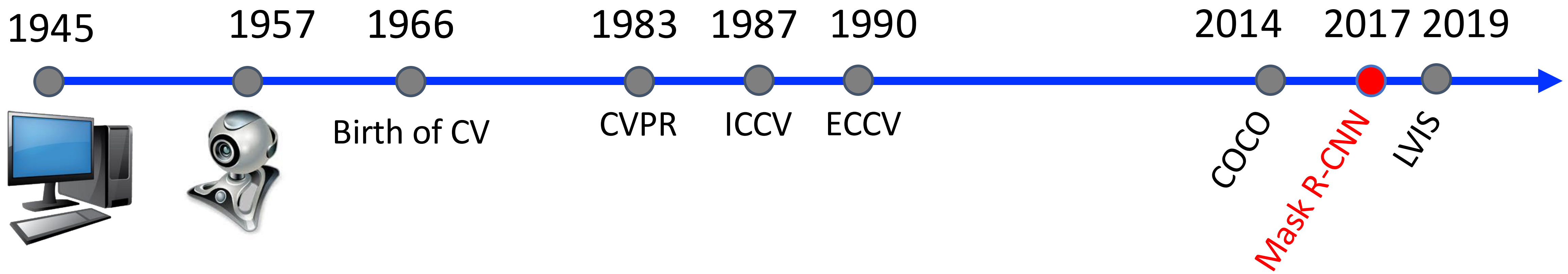
AP@[0.5:0.05:0.95]

- Average mAP when using multiple IoU thresholds to determine if a prediction matches a ground truth detection
  - 10 IoU thresholds, from 0.5 to 0.95 with a step size of 0.05

Instance Segmentation: Today’s Topics

- Motivation
- Datasets
- Evaluation metric
- Mask R-CNN
- YOLACT

Historical Context



Why Mask R-CNN?

Named after the approach of adapting Faster R-CNN to also predict **masks**:

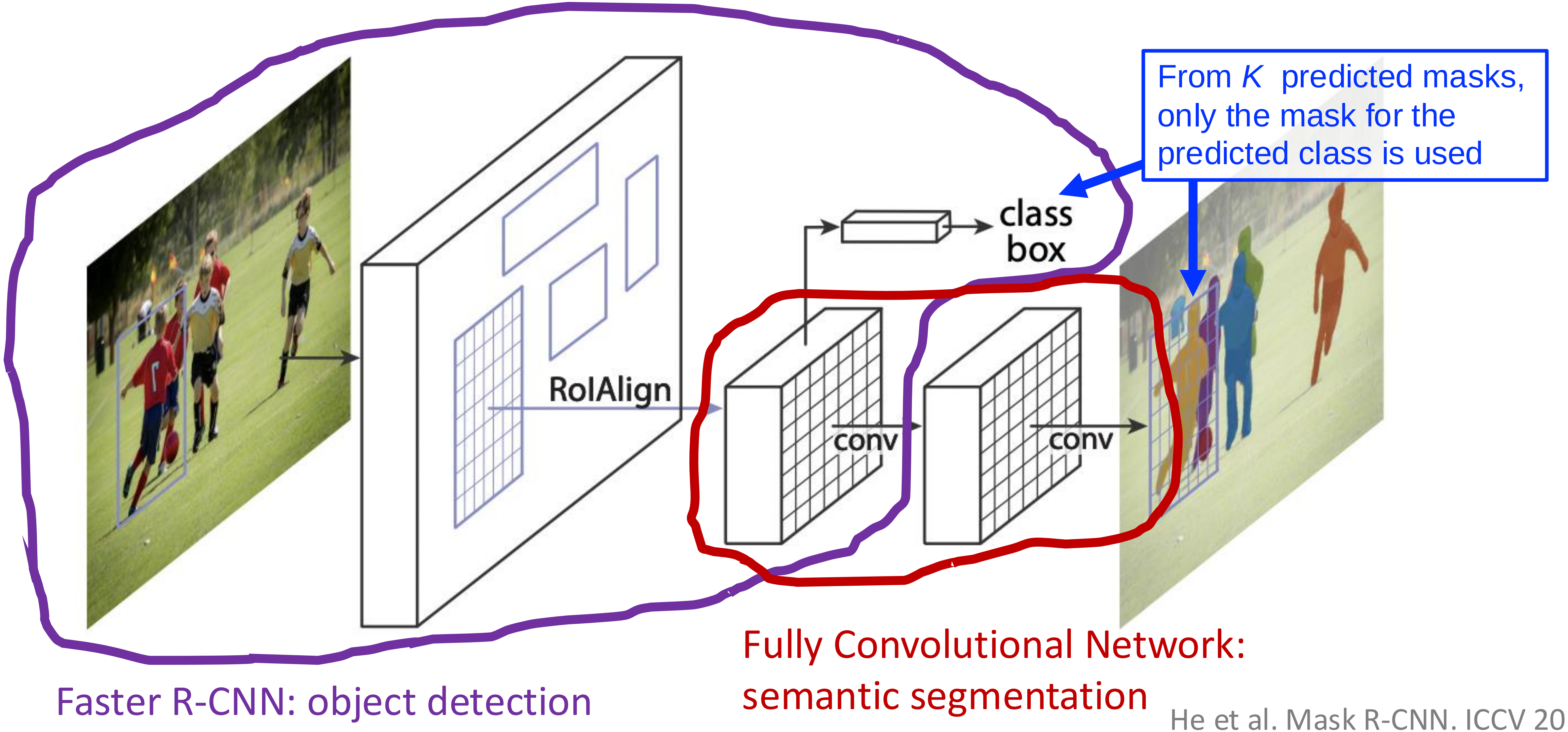
Kaiming He, Georgia Gkioxari, Piotr Dollar, & Ross Girshick. “Mask R-CNN.” ICCV 2017.

Key Contributions of Mask R-CNN

1. A pooling method that preserves the pixel-to-pixel alignment between the model’s input and output when downsampling

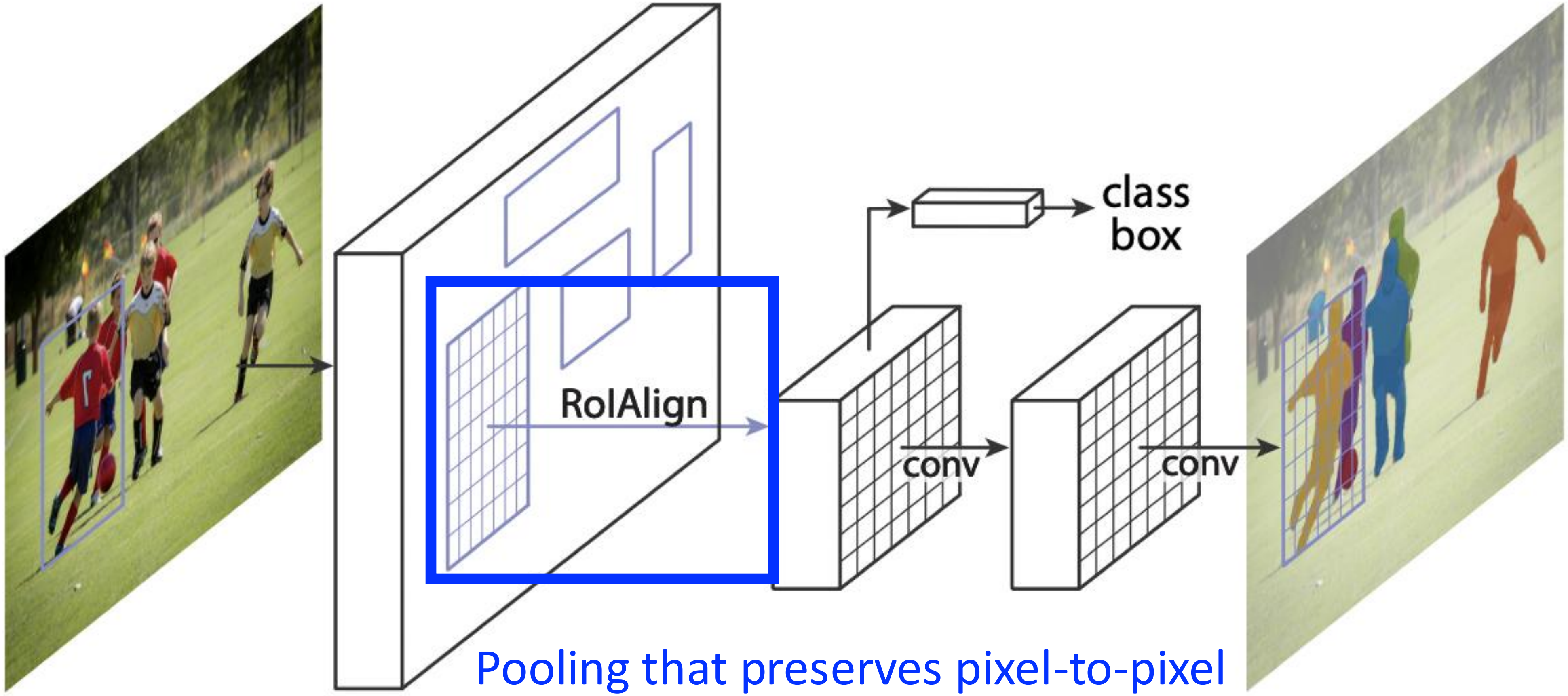
2. State-of-the-art performance on COCO

Architecture: Extends Faster R-CNN by Also Predicting in Parallel a Mask Per Region



He et al. Mask R-CNN. ICCV 2017

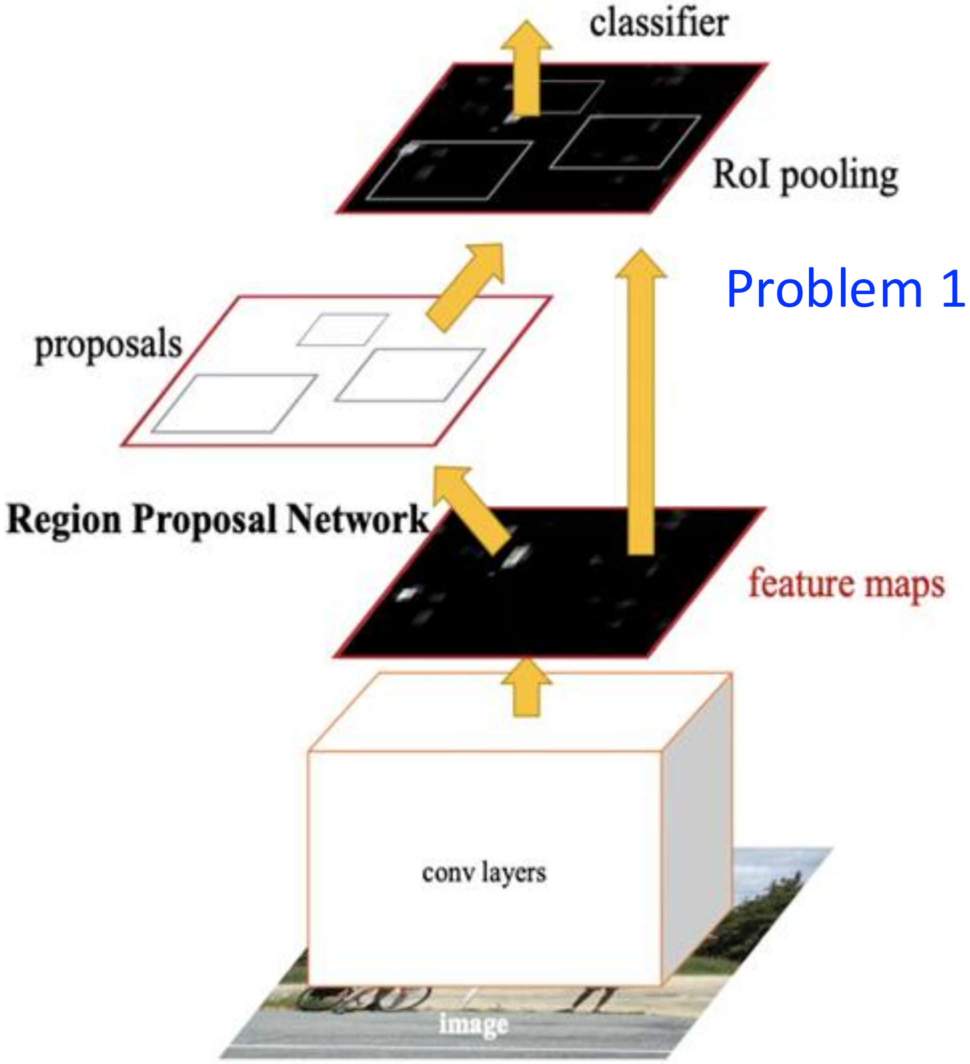
Architecture: Key Idea



Pooling that preserves pixel-to-pixel alignment between model's input and output

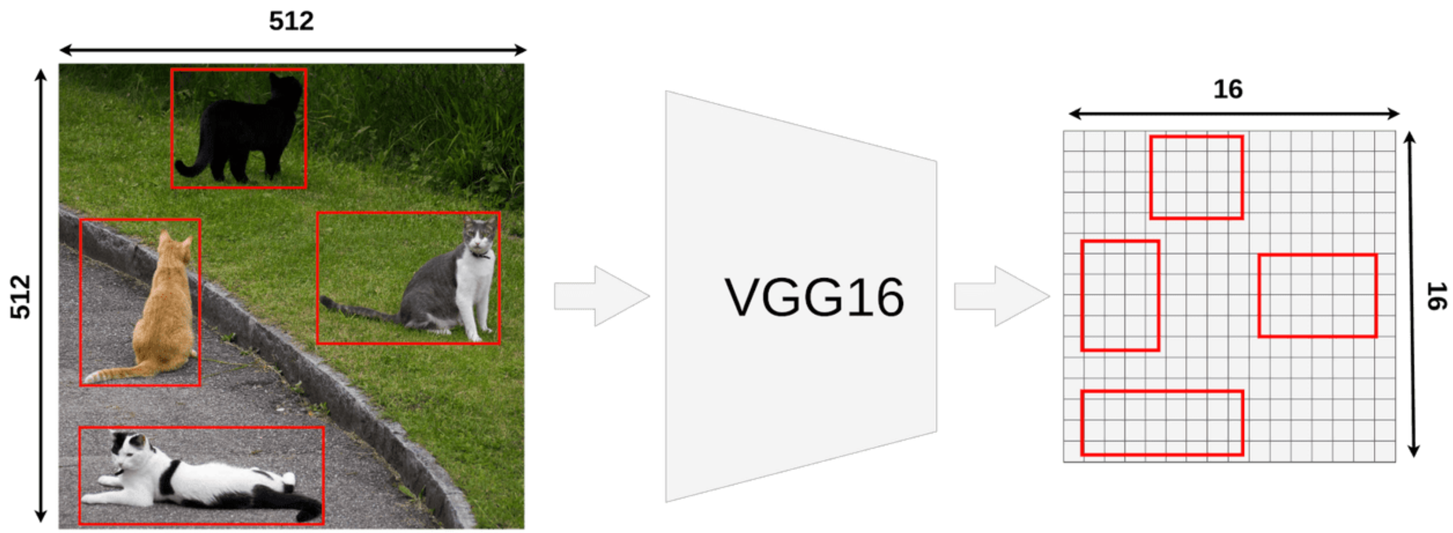
He et al. Mask R-CNN. ICCV 2017

ROIAlign Motivation: Revisiting Faster R-CNN



Problem 1: quantization of region proposals in a downsized feature map

e.g., 1/32 of the size (512/32 = 16)



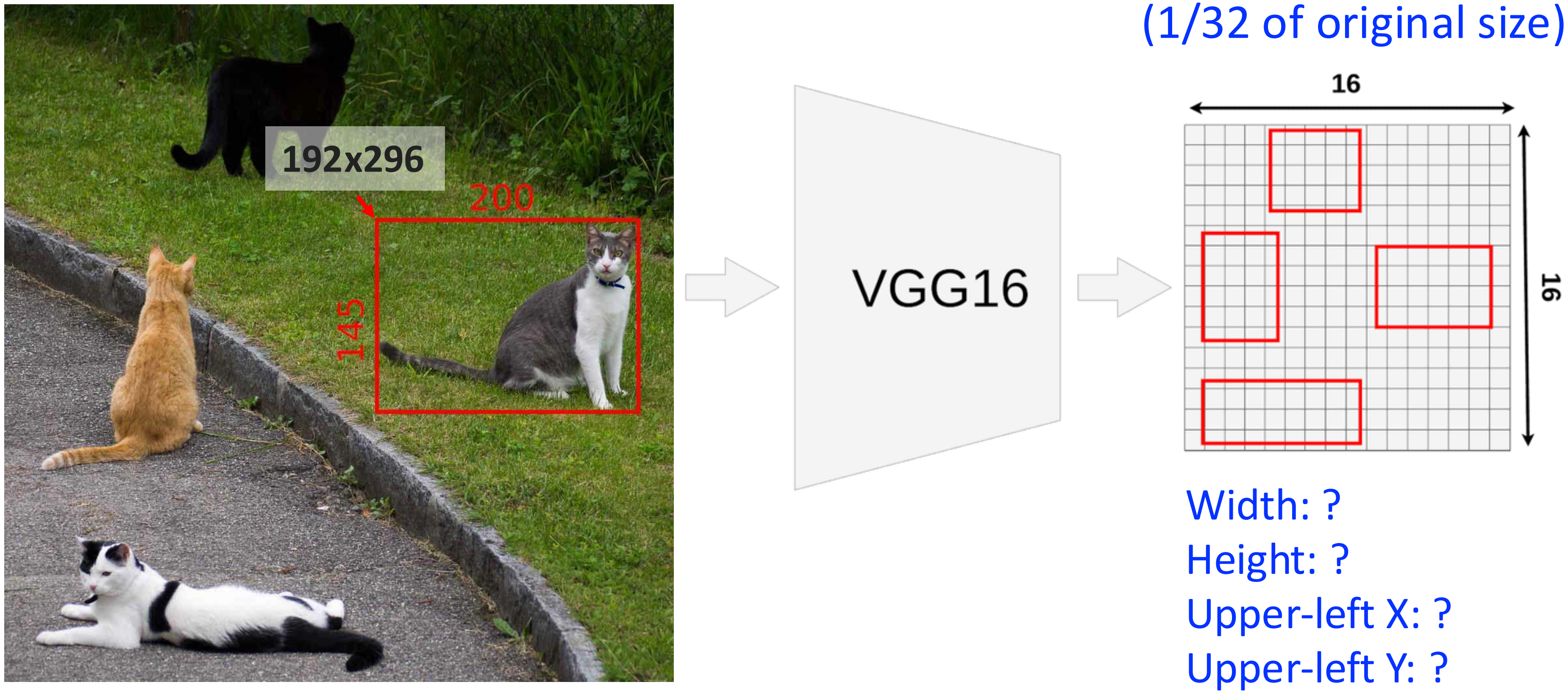
https://erdem.pl/2020/02/understanding-region-of-interest-ro-i-pooling

Ren Shaoqing Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." Neurips 2015

ROIAlign Motivation: Revisiting Faster R-CNN

What are the values for the region in the original image in the downsampled feature map?

(1/32 of original size)

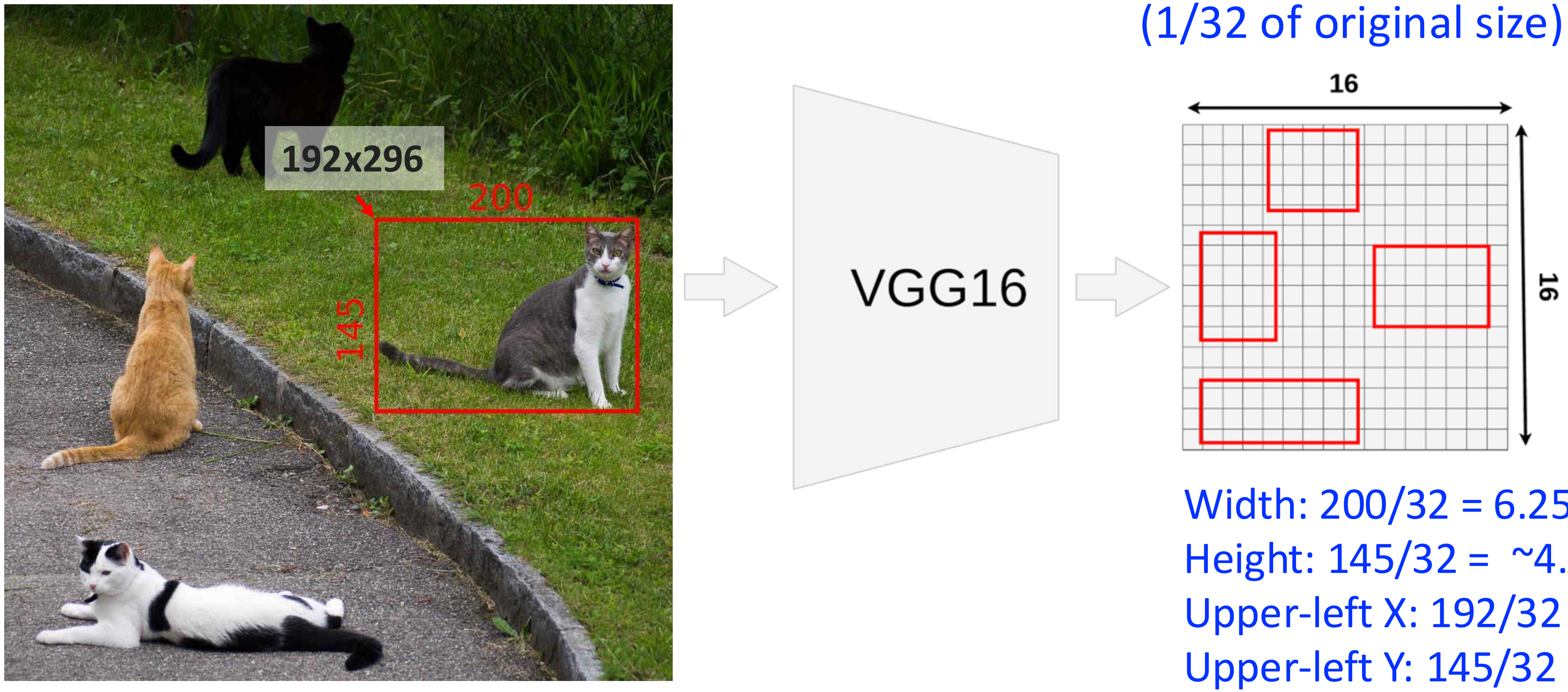


Width: ?
Height: ?
Upper-left X: ?
Upper-left Y: ?

https://erdem.pl/2020/02/understanding-region-of-interest-ro-i-pooling

ROIAlign Motivation: Revisiting Faster R-CNN

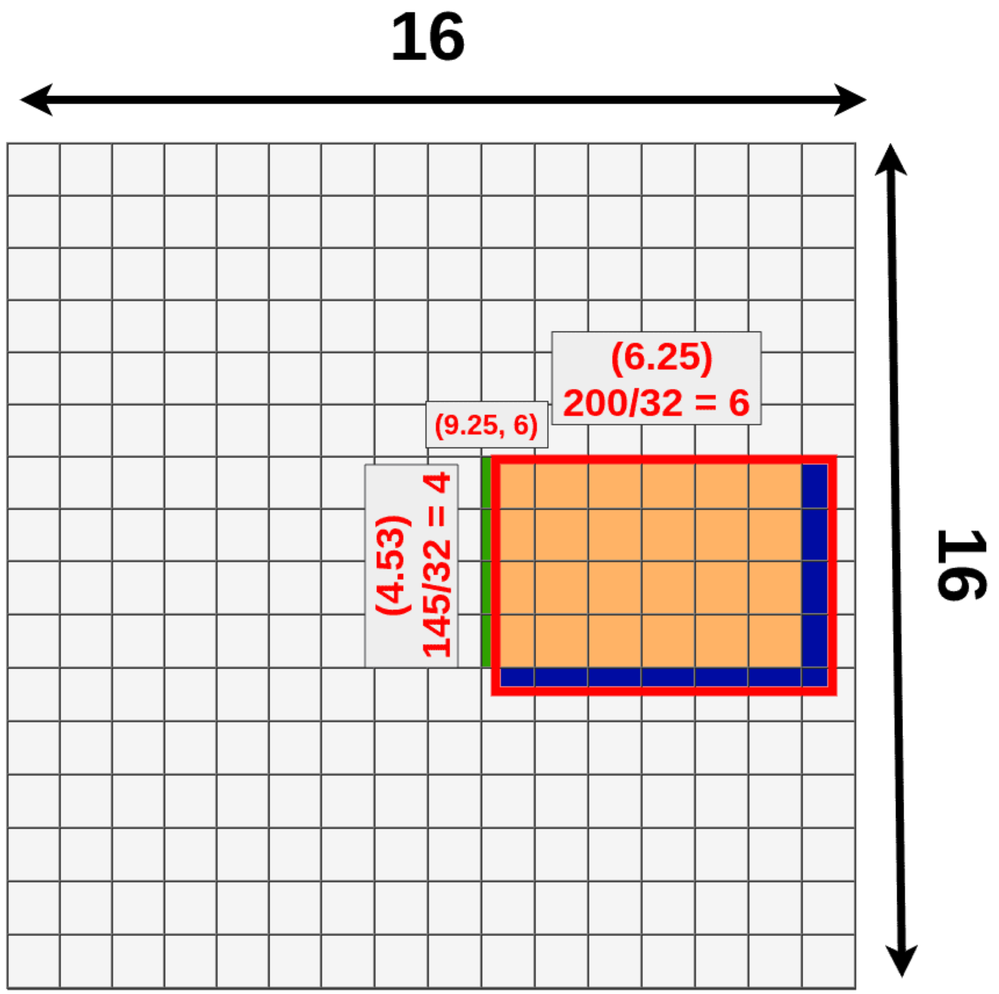
What are the values for the region in the original image in the downsampled feature map?



Width: 200/32 = 6.25
Height: 145/32 = ~4.53
Upper-left X: 192/32 = 9.25
Upper-left Y: 145/32 = 6

https://erdem.pl/2020/02/understanding-region-of-interest-ro-i-pooling

ROIAlign Motivation: Revisiting Faster R-CNN



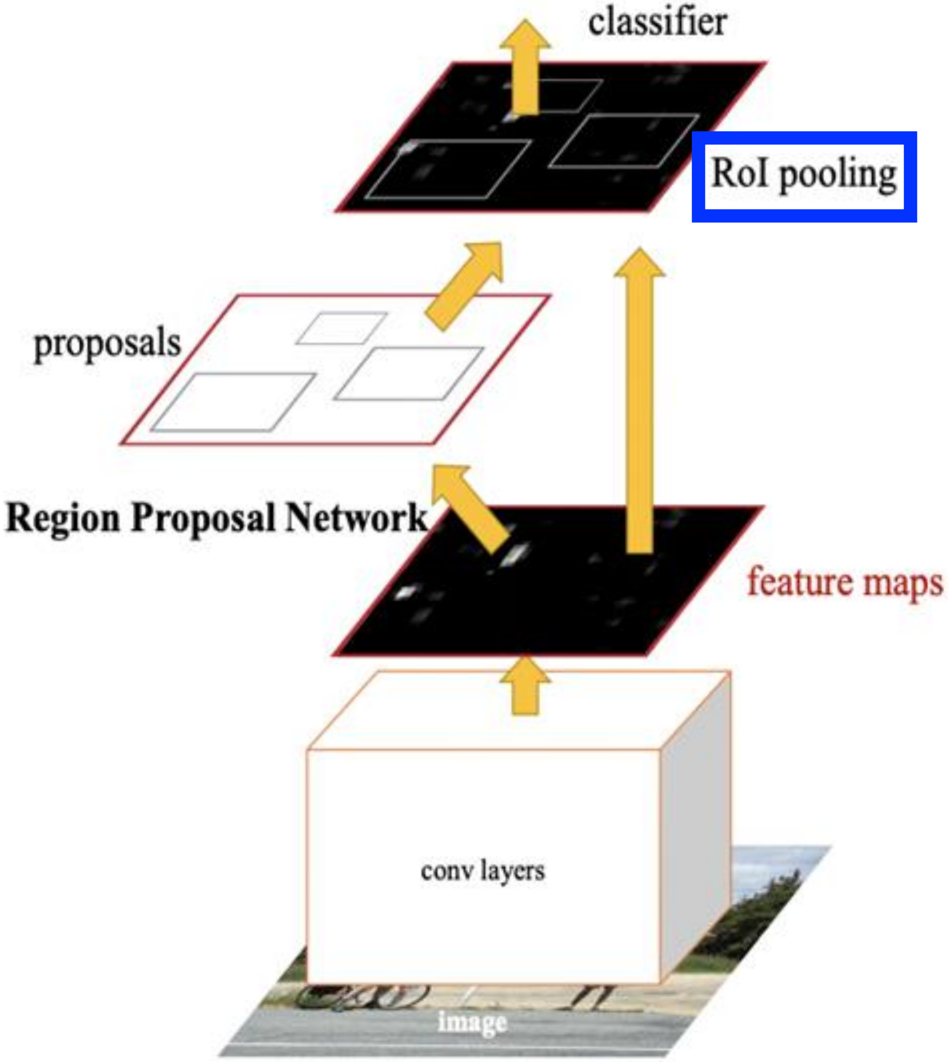
Original region on feature map

Quantized variant: values rounded down to only include a discrete set of integers to match the grid
- Original information preserved
- Information added
- Information lost

Quantization changes the information utilized from the original image, losing information about the object and adding extra image context (recall, the original image is orders of magnitude larger than the feature map!)

https://erdem.pl/2020/02/understanding-region-of-interest-ro-i-pooling

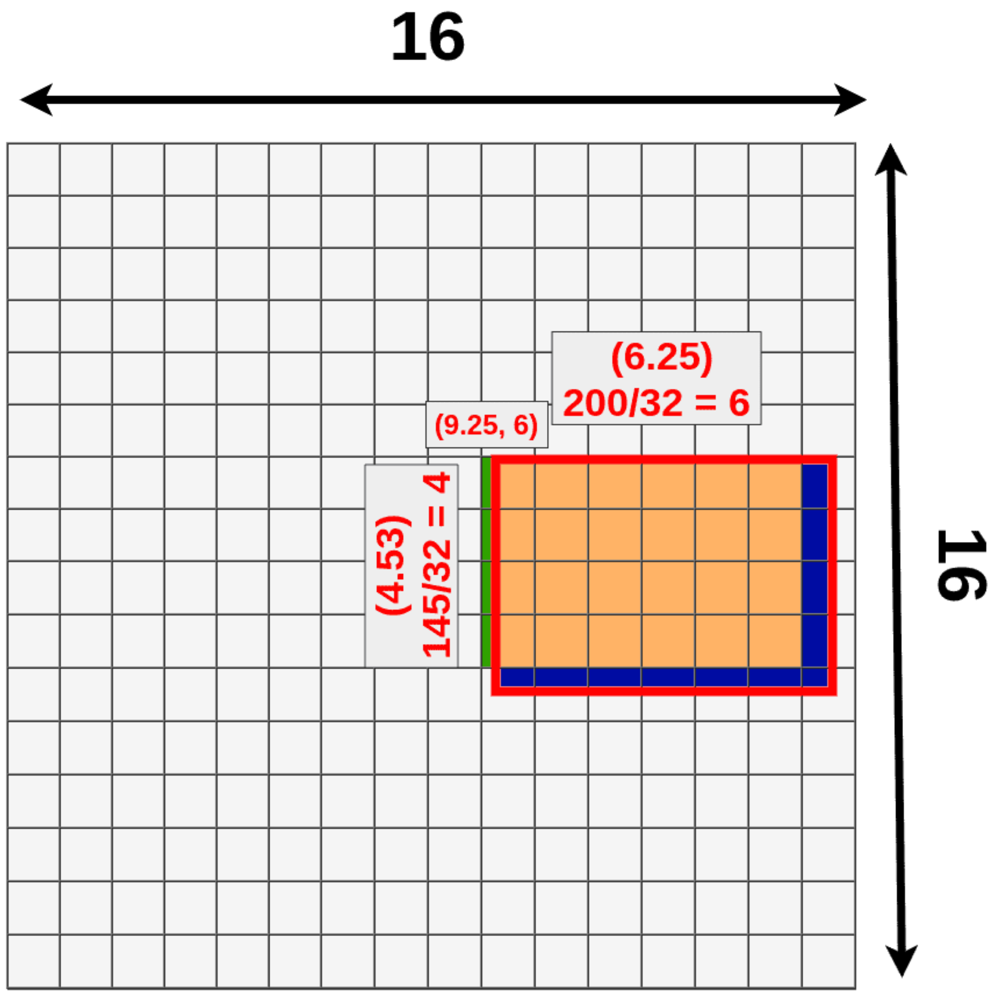
ROIAlign Motivation: Revisiting Faster R-CNN



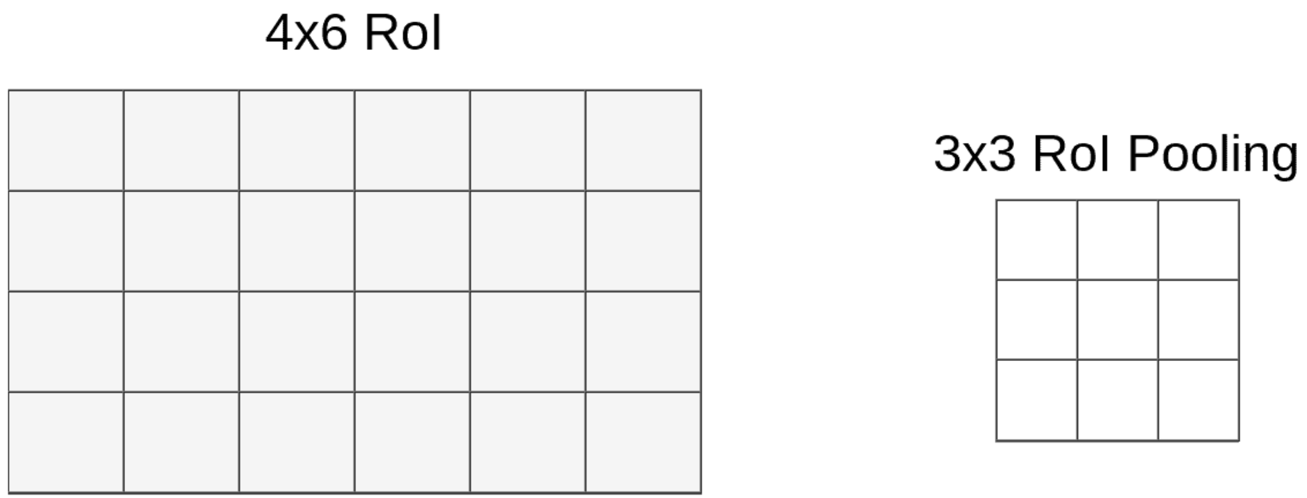
Problem 2: Quantization when pooling region proposals of various sizes to the fixed size required by the fully connected layer

Ren Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” Neurips 2015

ROIAlign Motivation: Revisiting Faster R-CNN



e.g., convert quantized 4x6 region into a 3x3 feature

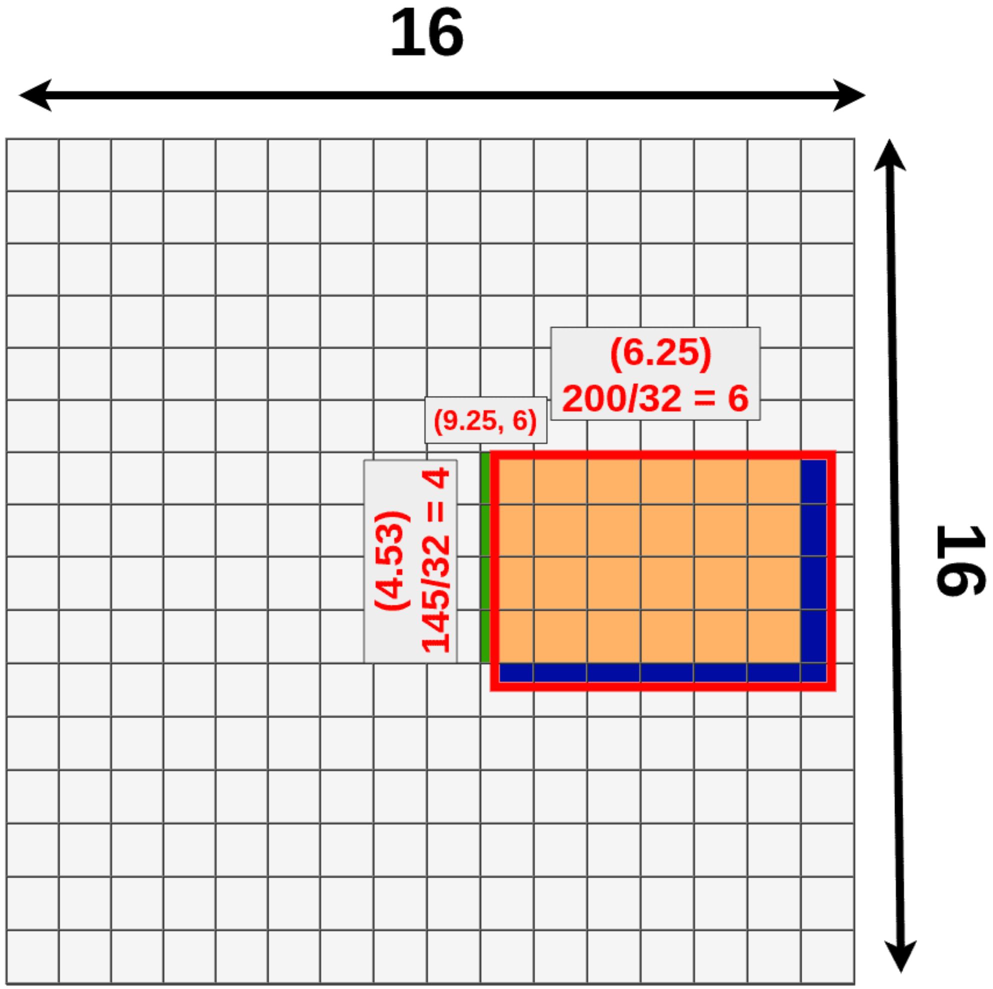


Quantized approach: identify discrete integers for pooling to result in the target size

e.g., 4/3 = 1.3 -> 1 and 6/3 = 2

https://erdem.pl/2020/02/understanding-region-of-interest-ro-i-pooling

ROIAlign Motivation: Revisiting Faster R-CNN

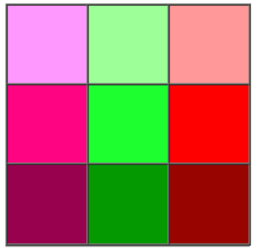


e.g., convert quantized 4x6 region into a 3x3 feature

4x6 RoI

0.1	0.2	0.3	0.4	0.5	0.6
1	0.7	0.2	0.6	0.1	0.9
0.9	0.8	0.7	0.3	0.5	0.2

3x3 RoI Pooling

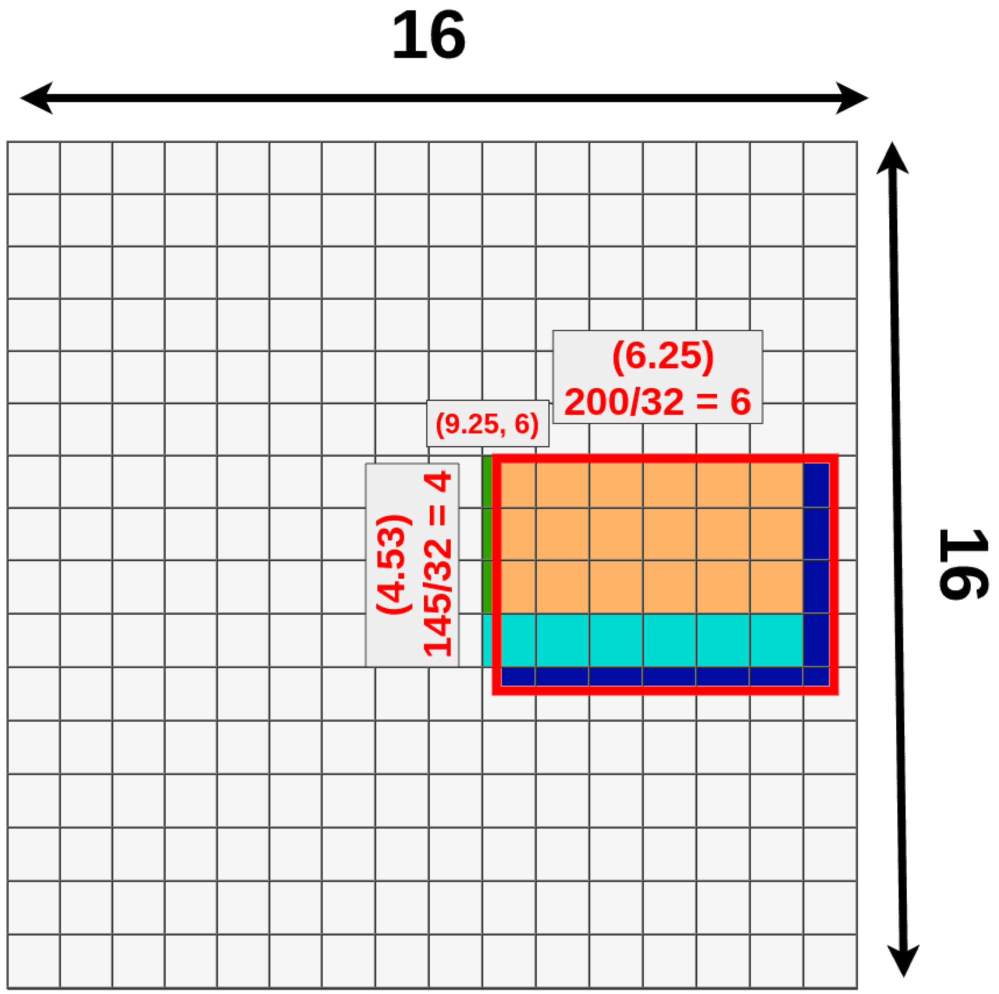


Quantized approach: identify discrete integers for pooling to result in the target size

e.g., 1x2 vector using max pooling

https://erdem.pl/2020/02/understanding-region-of-interest-ro-i-pooling

ROIAlign Motivation: Revisiting Faster R-CNN



e.g., convert quantized 4x6 region into a 3x3 feature

4x6 RoI

0.1	0.2	0.3	0.4	0.5	0.6
1	0.7	0.2	0.6	0.1	0.9
0.9	0.8	0.7	0.3	0.5	0.2
0.2	0.5	1	0.7	0.1	0.1

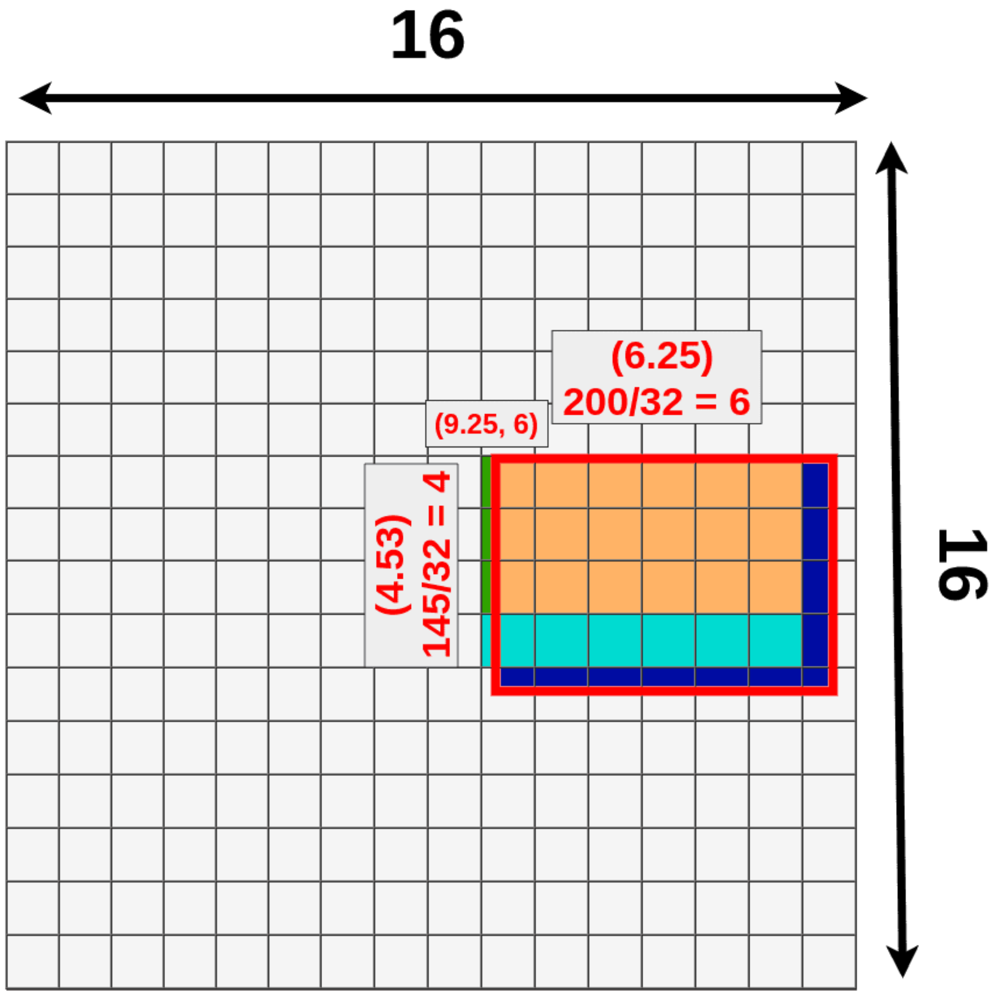
Again, quantization discards information about the object from the original image (recall, the original image is orders of magnitude larger than the feature map!)

Quantized approach: identify discrete integers for pooling to result in the target size

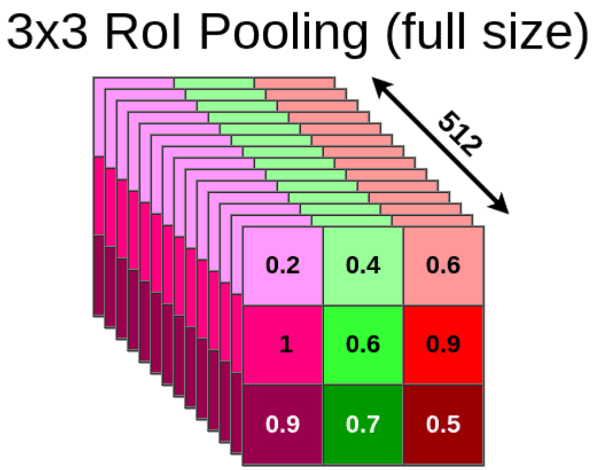
e.g., 1x2 vector using max pooling

https://erdem.pl/2020/02/understanding-region-of-interest-ro-i-pooling

ROIAlign Motivation: Revisiting Faster R-CNN



e.g., convert quantized 4x6 region into a 3x3 feature



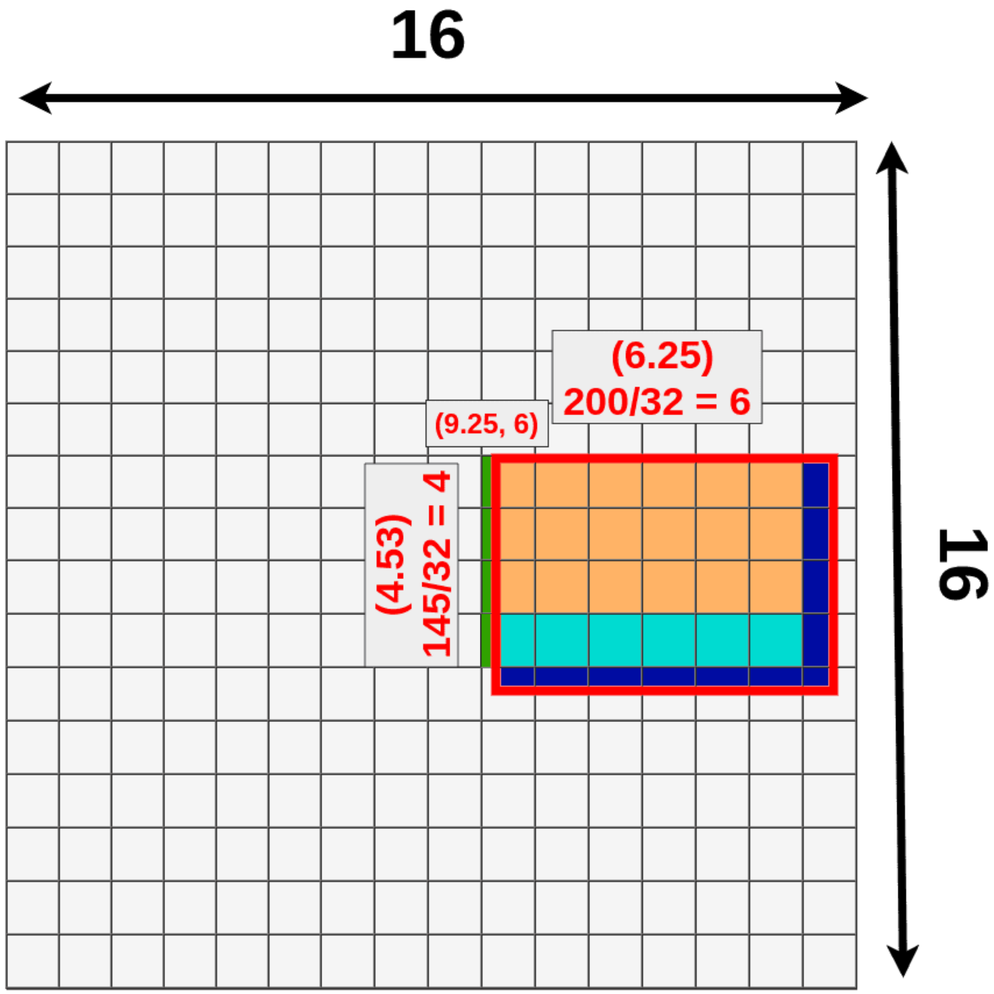
Information is lost for *all* channels for *every* region proposal (each of which is used to predict a class and bounding box)!

Quantized approach: identify discrete integers for pooling to result in the target size

e.g., 1x2 vector using max pooling

https://erdem.pl/2020/02/understanding-region-of-interest-ro-i-pooling

ROIAlign Motivation: Summary

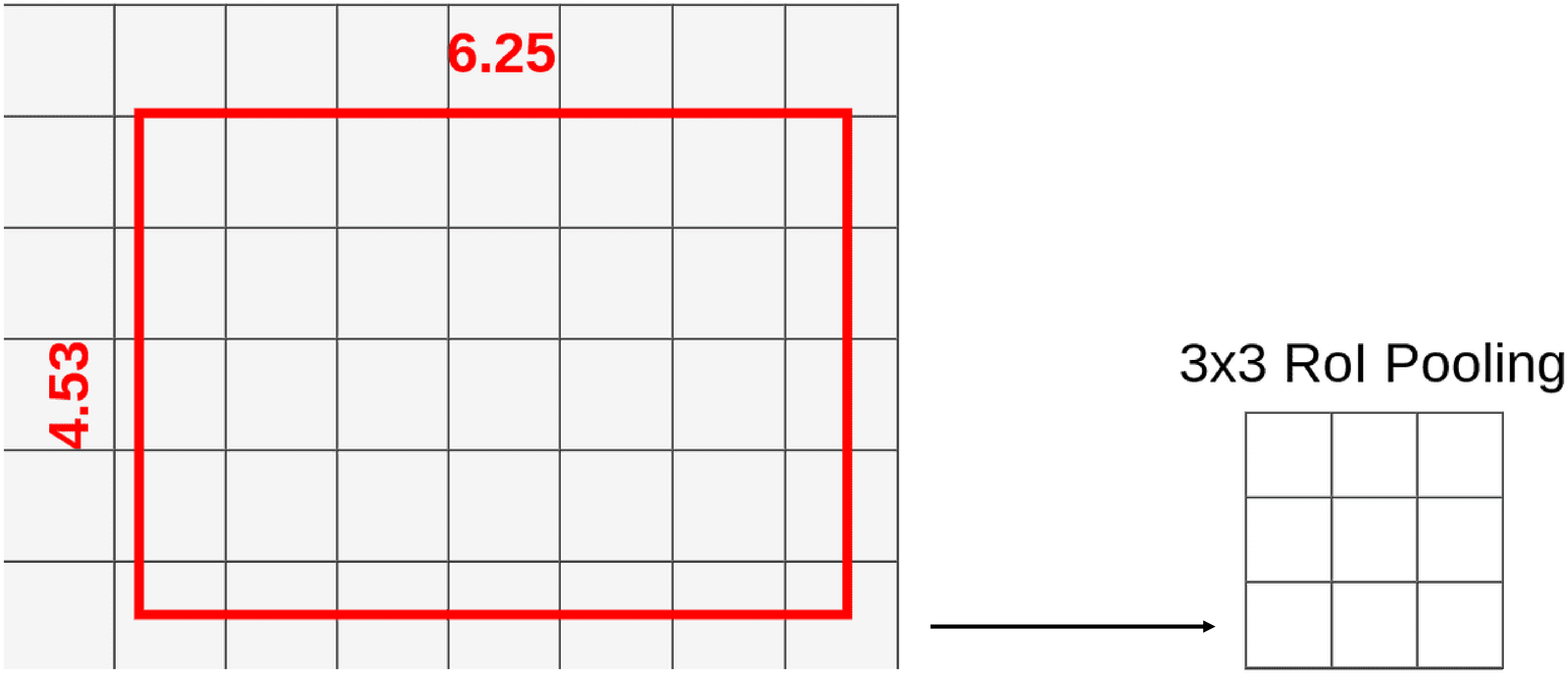


Original region on feature map

Quantization changes the information utilized from the original image, losing information about the object and adding extra image context (recall, the original image is orders of magnitude larger than the feature map!)

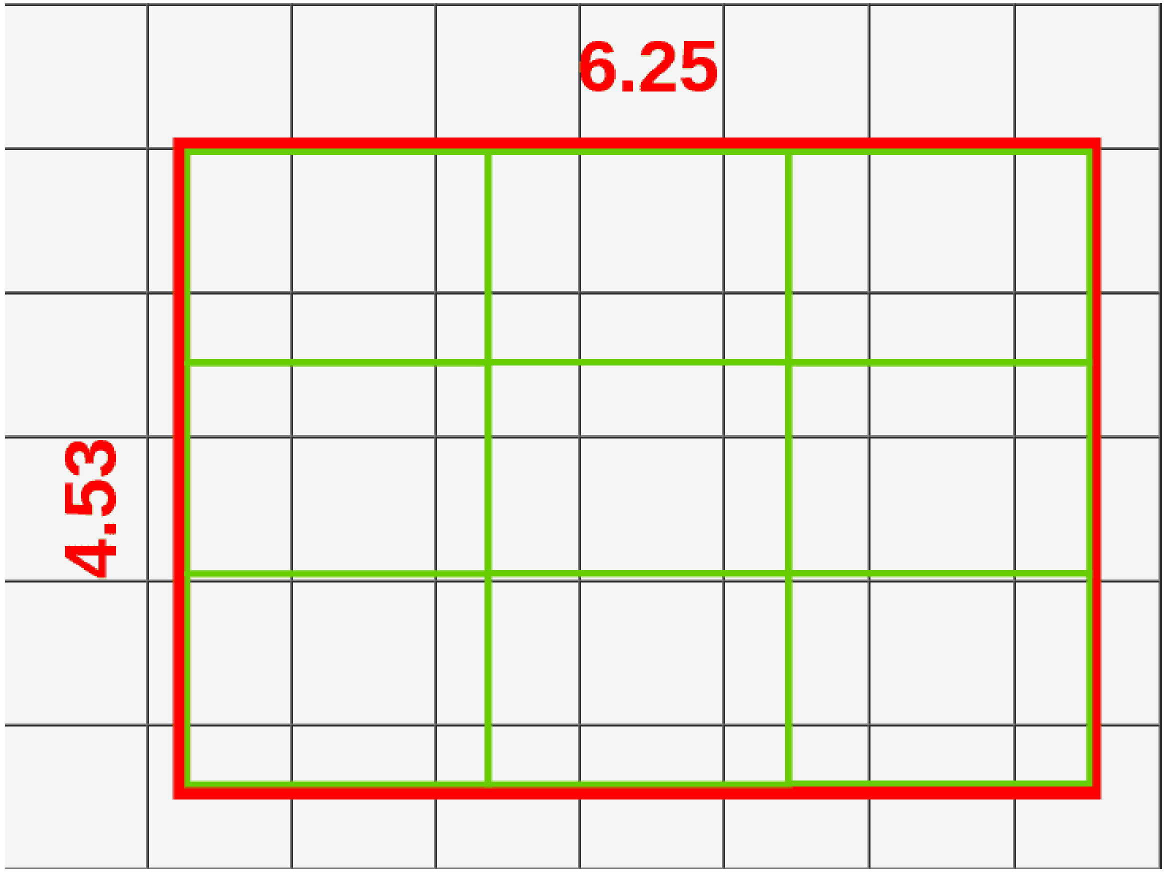
https://erdem.pl/2020/02/understanding-region-of-interest-ro-i-pooling

ROIAlign: Pooling *Without* Quantization



https://erdem.pl/2020/02/understanding-region-of-interest-part-2-ro-i-align

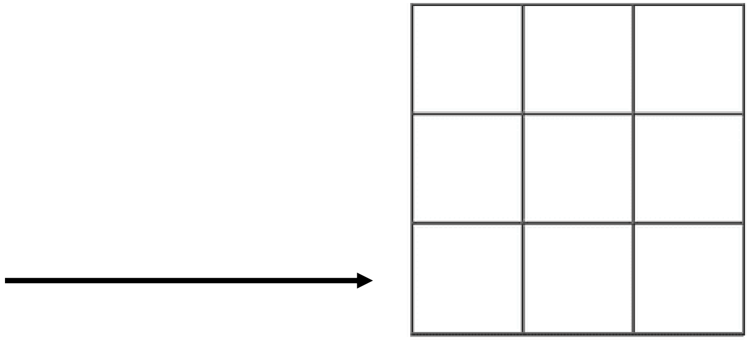
ROIAlign: Pooling ***Without*** Quantization



Divide region into 9 equal sized boxes; what is the size of each box?

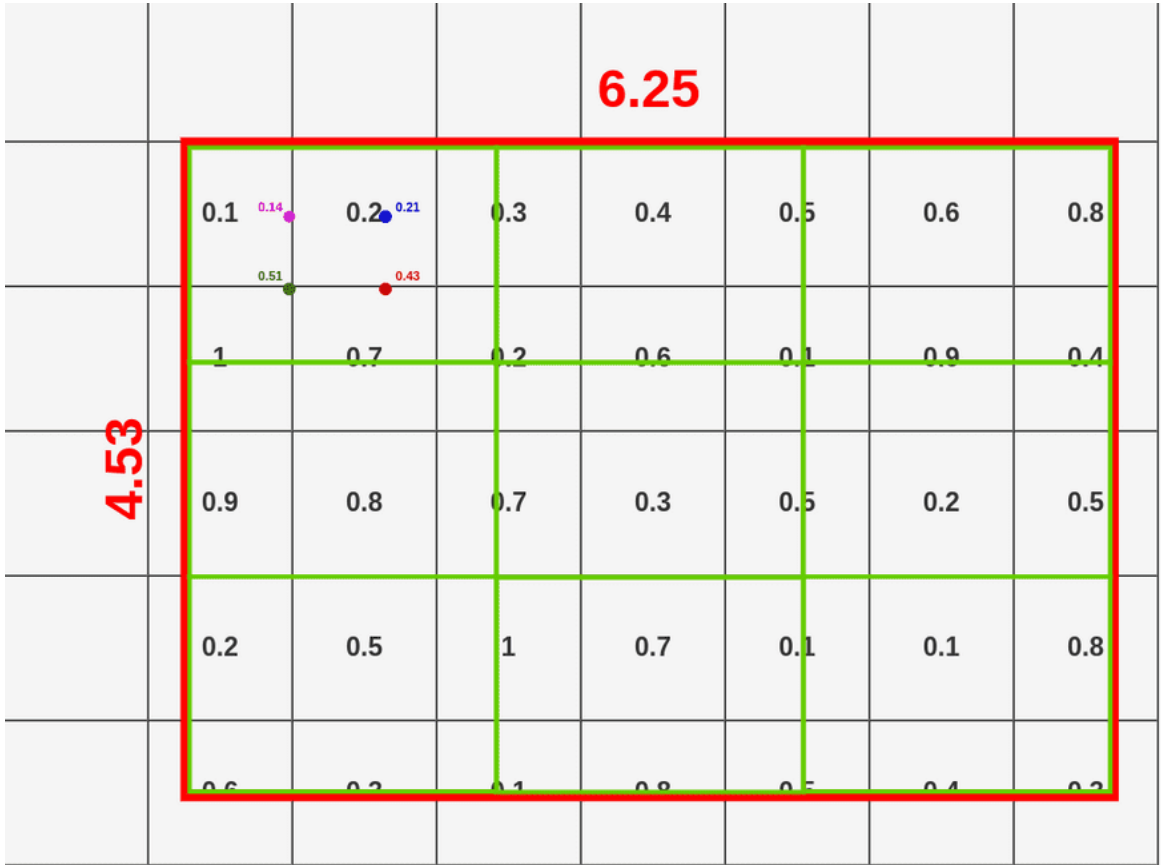
- 6.25/3 x 4.53/3 = 2.08 x 1.51

3x3 RoI Pooling



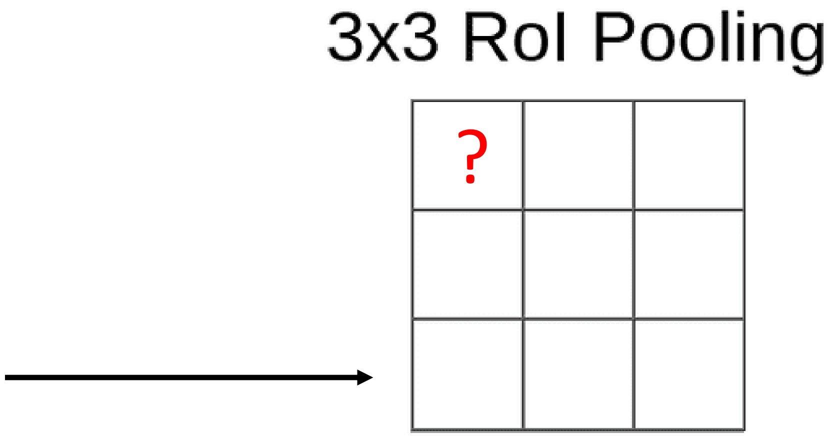
https://erdem.pl/2020/02/understanding-region-of-interest-part-2-ro-i-align

ROIAlign: Pooling *Without* Quantization



Perform pooling on sampled values in each box
- e.g., max(0.14, 0.21, 0.51, 0.43) = ?

How do we find the four sample values?



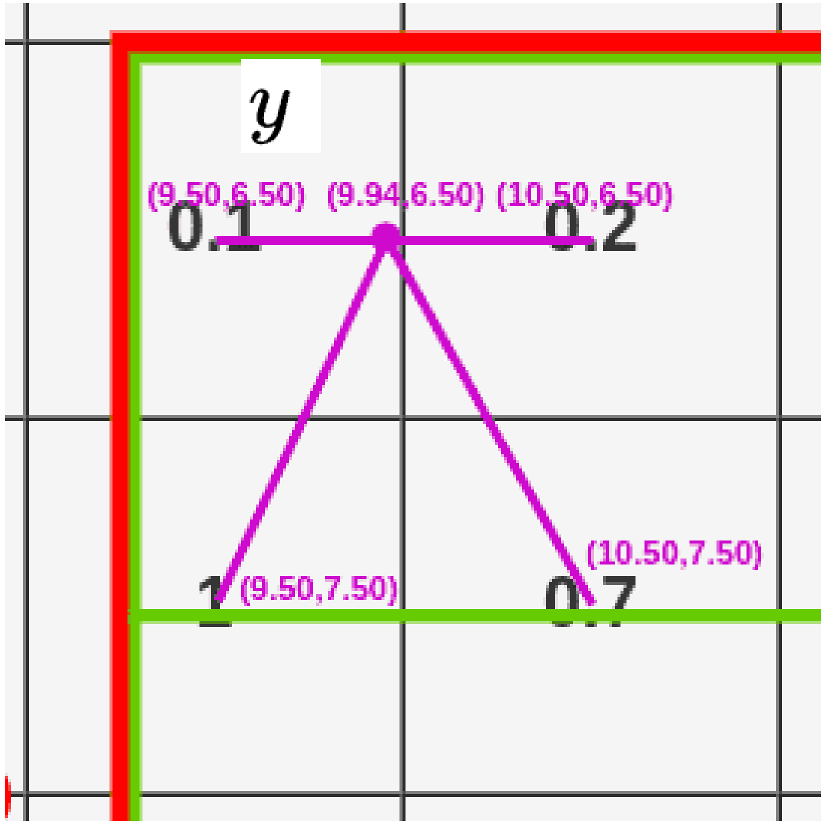
https://erdem.pl/2020/02/understanding-region-of-interest-part-2-ro-i-align

ROIAlign: Pooling *Without* Quantization

Compute each sample value with interpolation between 4 points

https://erdem.pl/2020/02/understanding-region-of-interest-part-2-ro-i-align

ROIAlign: Pooling *Without* Quantization



Compute each **sample value** with interpolation between 4 points:

1. Identify sample location

$$y \begin{cases} X = X\_box + (width/3) * 1 = 9.25 + (2.08/3) = 9.94 \\ Y = Y\_box + (height/3) * 1 = 6 + (1.51/3) = 6.50 \end{cases}$$

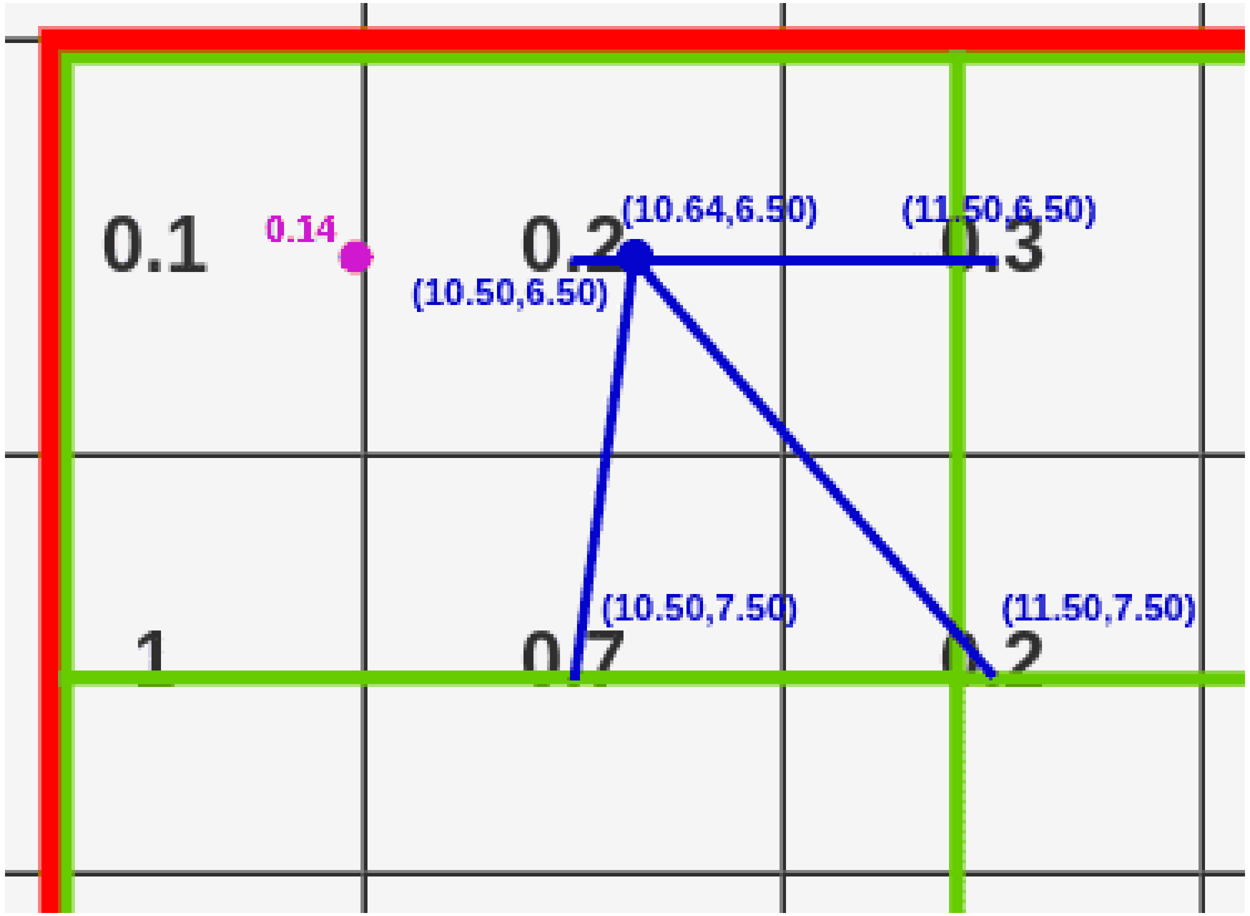
2. Identify 4 points for interpolation, using the middle of each closest neighboring box in each direction
3. Calculate value using bilinear interpolation (= 0.14)

$$P \approx \frac{y_2 - y}{y_2 - y_1}\left(\frac{x_2 - x}{x_2 - x_1}Q_{11} + \frac{x - x_1}{x_2 - x_1}Q_{21}\right) + \frac{y - y_1}{y_2 - y_1}\left(\frac{x_2 - x}{x_2 - x_1}Q_{12} + \frac{x - x_1}{x_2 - x_1}Q_{22}\right)$$

$$\approx \frac{7.5 - 6.5}{7.5 - 6.5}\left(\frac{10.5 - 9.94}{10.5 - 9.5}0.1 + \frac{9.94 - 9.5}{10.5 - 9.5}0.2\right) + \frac{6.5 - 6.5}{7.5 - 6.5}\left(\frac{10.5 - 9.94}{10.5 - 9.5}1 + \frac{9.94 - 9.5}{10.5 - 9.5}0.7\right)$$

https://erdem.pl/2020/02/understanding-region-of-interest-part-2-ro-i-align

ROIAlign: Pooling *Without* Quantization

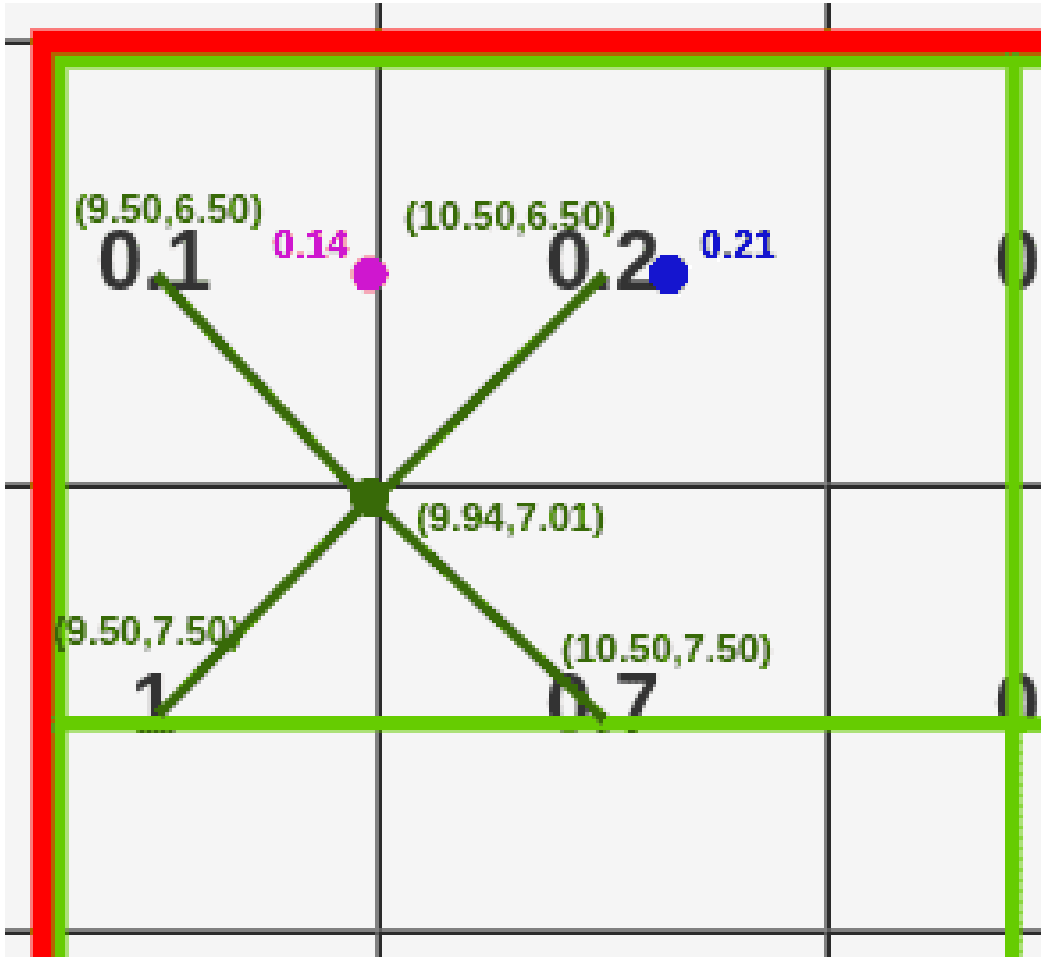


Compute each sample value with interpolation between 4 points:

1. Identify sample location
2. Identify 4 points for interpolation, using the middle of each closest neighboring box in each direction
3. Calculate value using bilinear interpolation (=0.21)

https://erdem.pl/2020/02/understanding-region-of-interest-part-2-ro-i-align

ROIAlign: Pooling *Without* Quantization

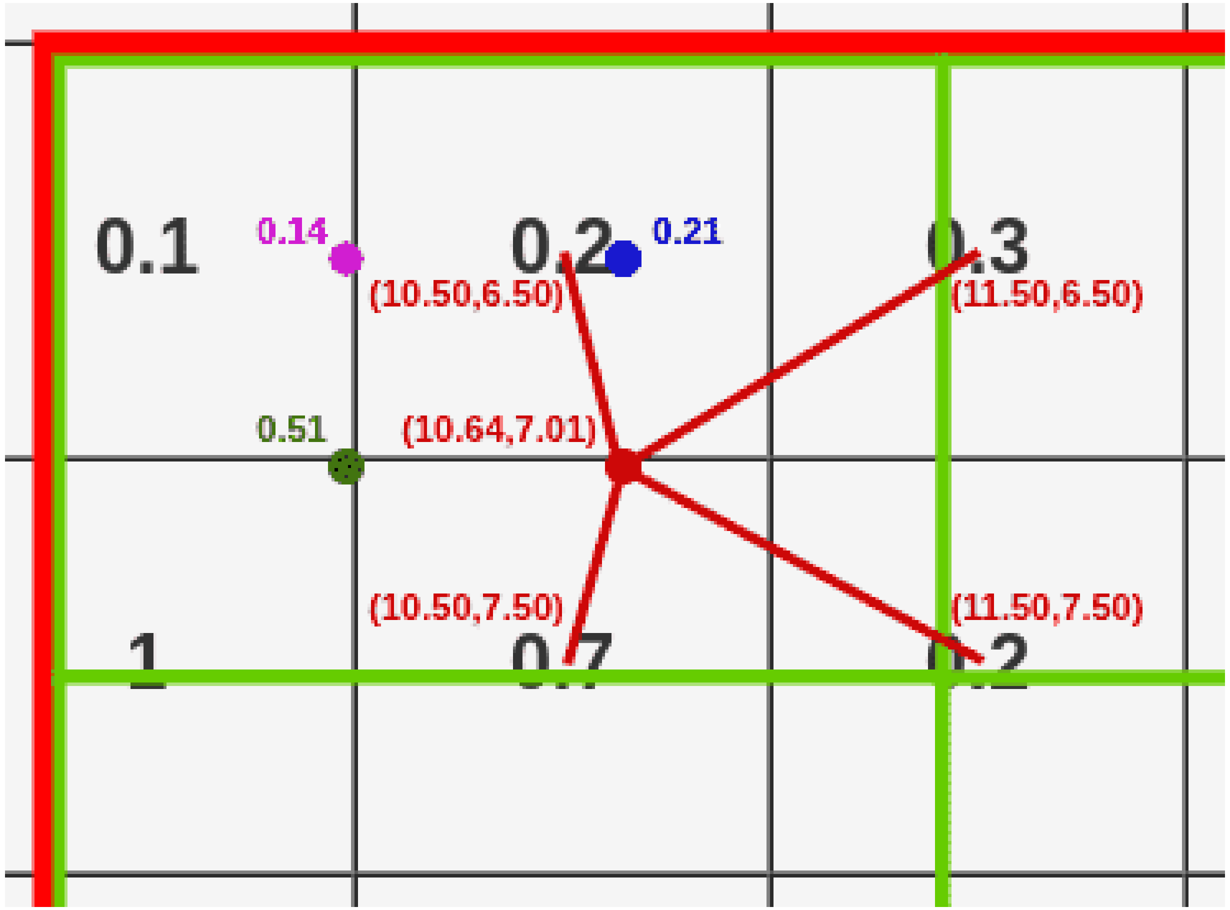


Compute each **sample value** with interpolation between 4 points:

1. Identify sample location
2. Identify 4 points for interpolation, using the middle of each closest neighboring box in each direction
3. Calculate value using bilinear interpolation (=0.51)

https://erdem.pl/2020/02/understanding-region-of-interest-part-2-ro-i-align

ROIAlign: Pooling *Without* Quantization

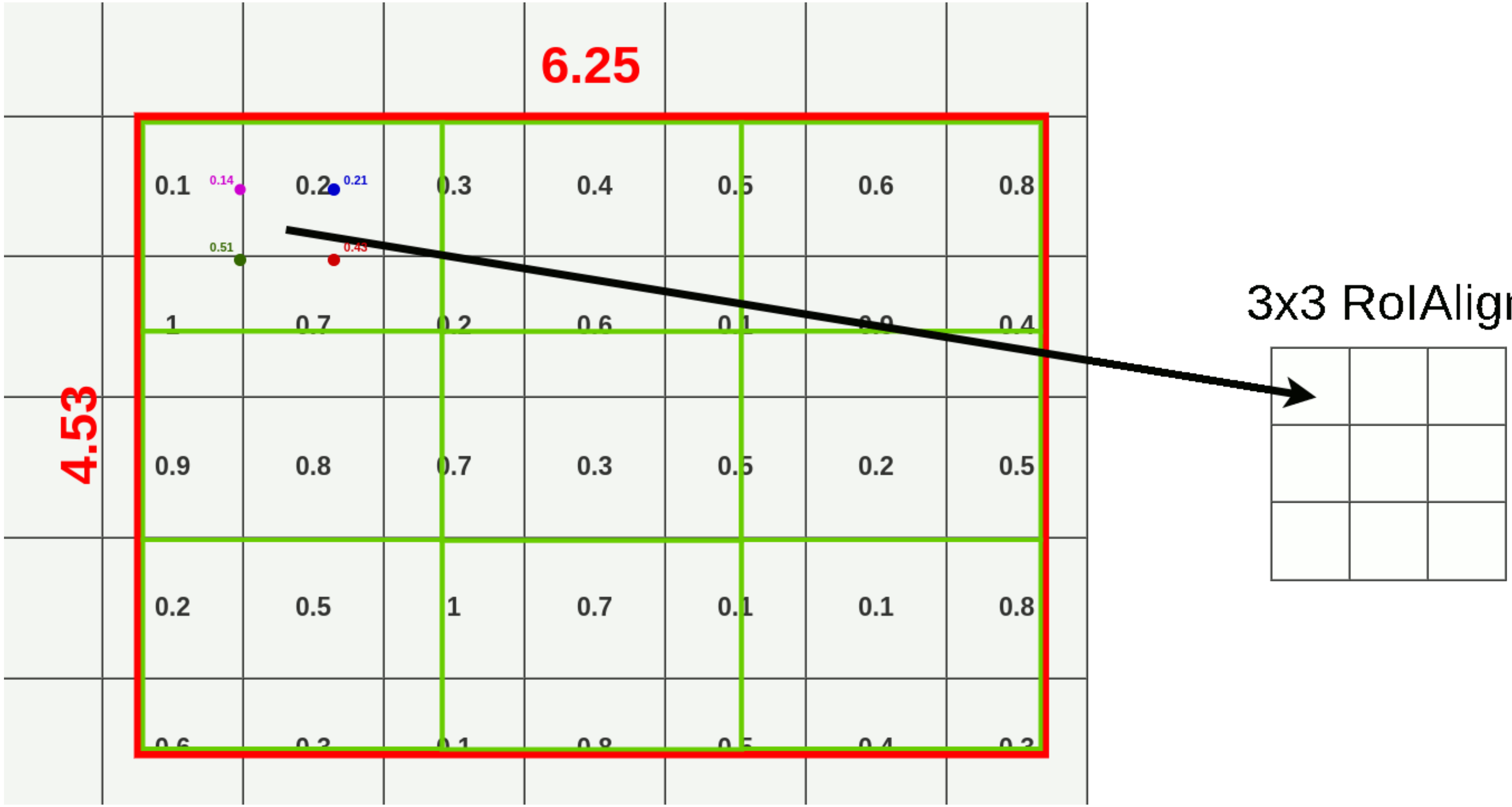


Compute each sample value with interpolation between 4 points:

1. Identify sample location
2. Identify 4 points for interpolation, using the middle of each closest neighboring box in each direction
3. Calculate value using bilinear interpolation (=0.43)

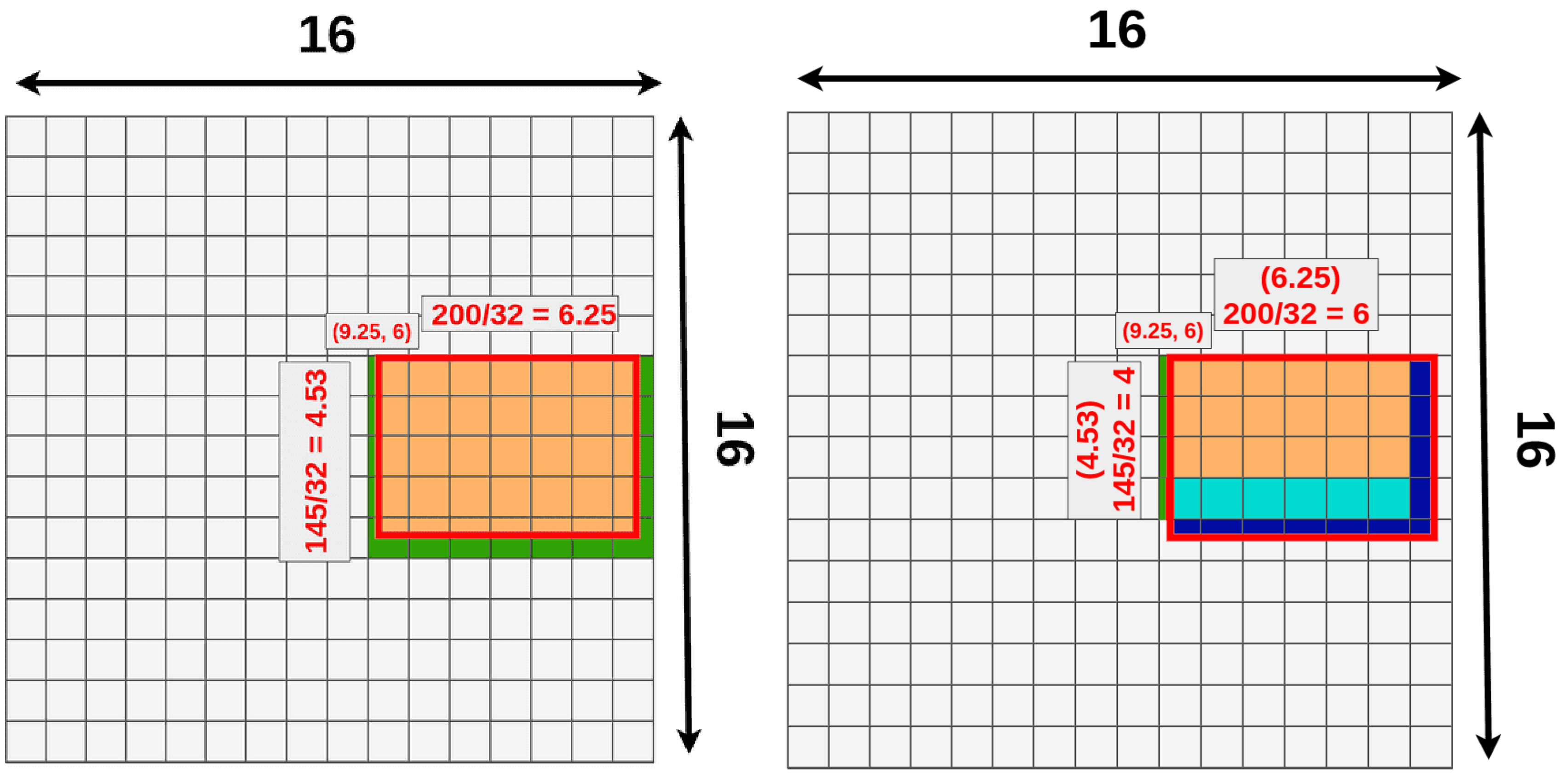
https://erdem.pl/2020/02/understanding-region-of-interest-part-2-ro-i-align

ROIAlign: Pooling *Without* Quantization



https://erdem.pl/2020/02/understanding-region-of-interest-part-2-ro-i-align

ROIAlign vs ROI Pooling



Original region on feature map

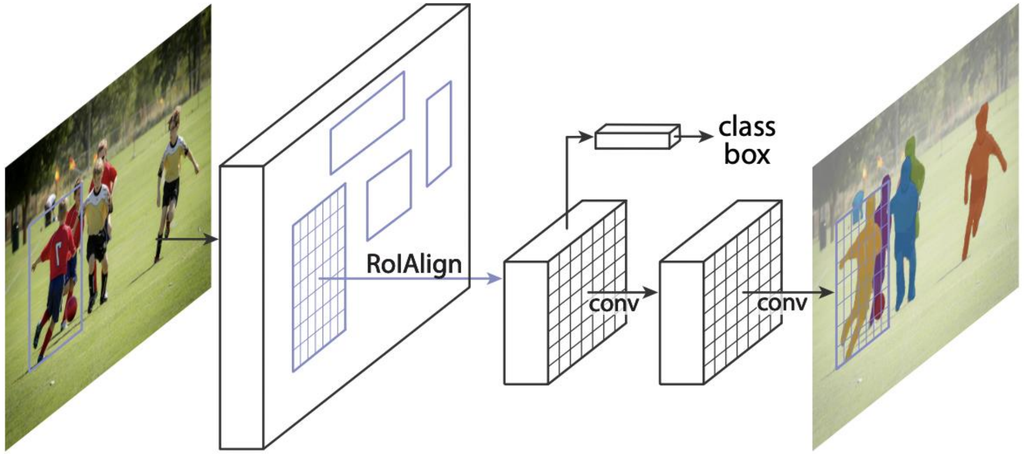
Both methods add extra image context

Only ROI pooling loses information about the object from the original image

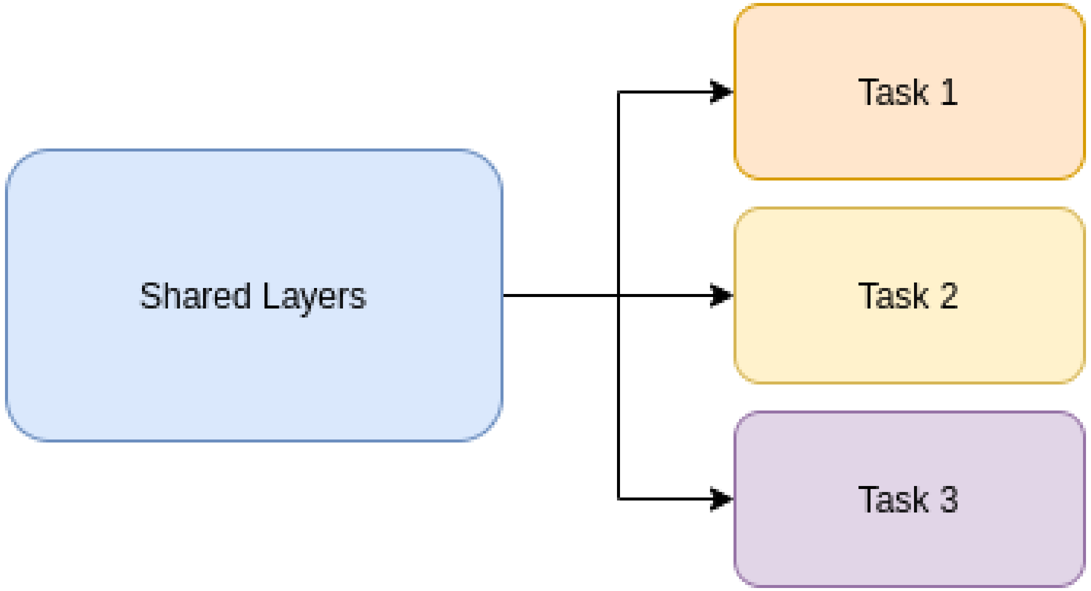
https://erdem.pl/2020/02/understanding-region-of-interest-part-2-ro-i-align

Training: Multi-Task Learning

What are the three tasks (and so types of losses) used during training?



He et al. Mask R-CNN. ICCV 2017

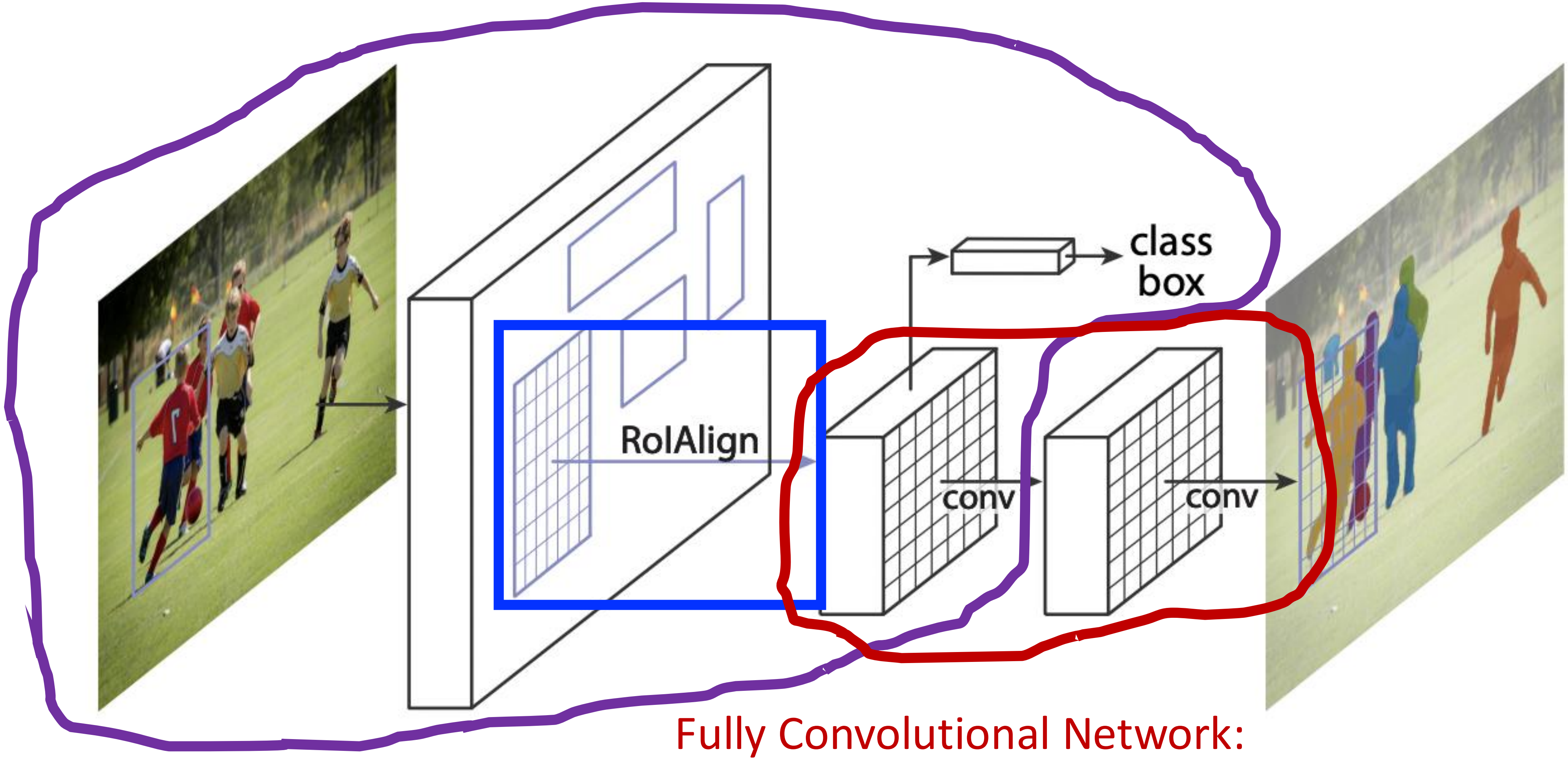


https://towardsdatascience.com/multi-task-learning-with-pytorch-and-fastai-6d10dc7ce855

$$L = L_{class} + L_{box} + L_{mask}$$

Summary: Focus for Today's Coding Tutorial

Faster R-CNN: object detection



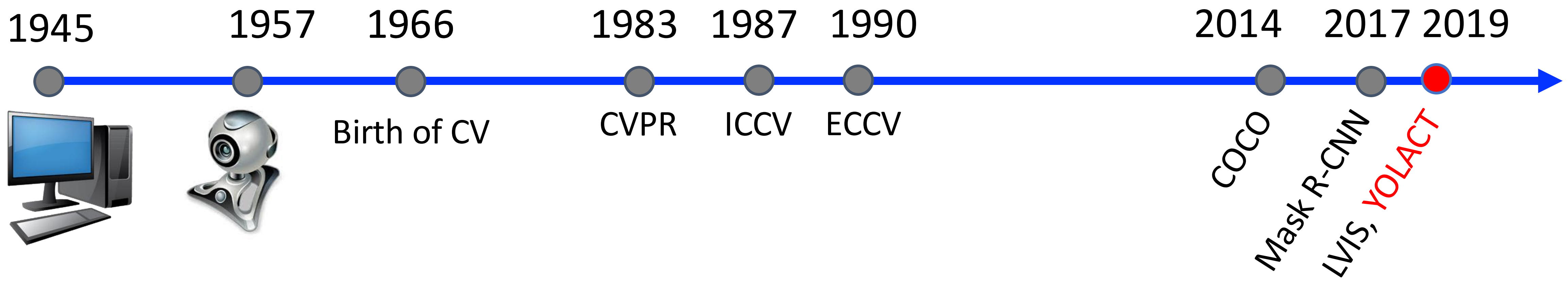
Fully Convolutional Network:
semantic segmentation

He et al. Mask R-CNN. ICCV 2017

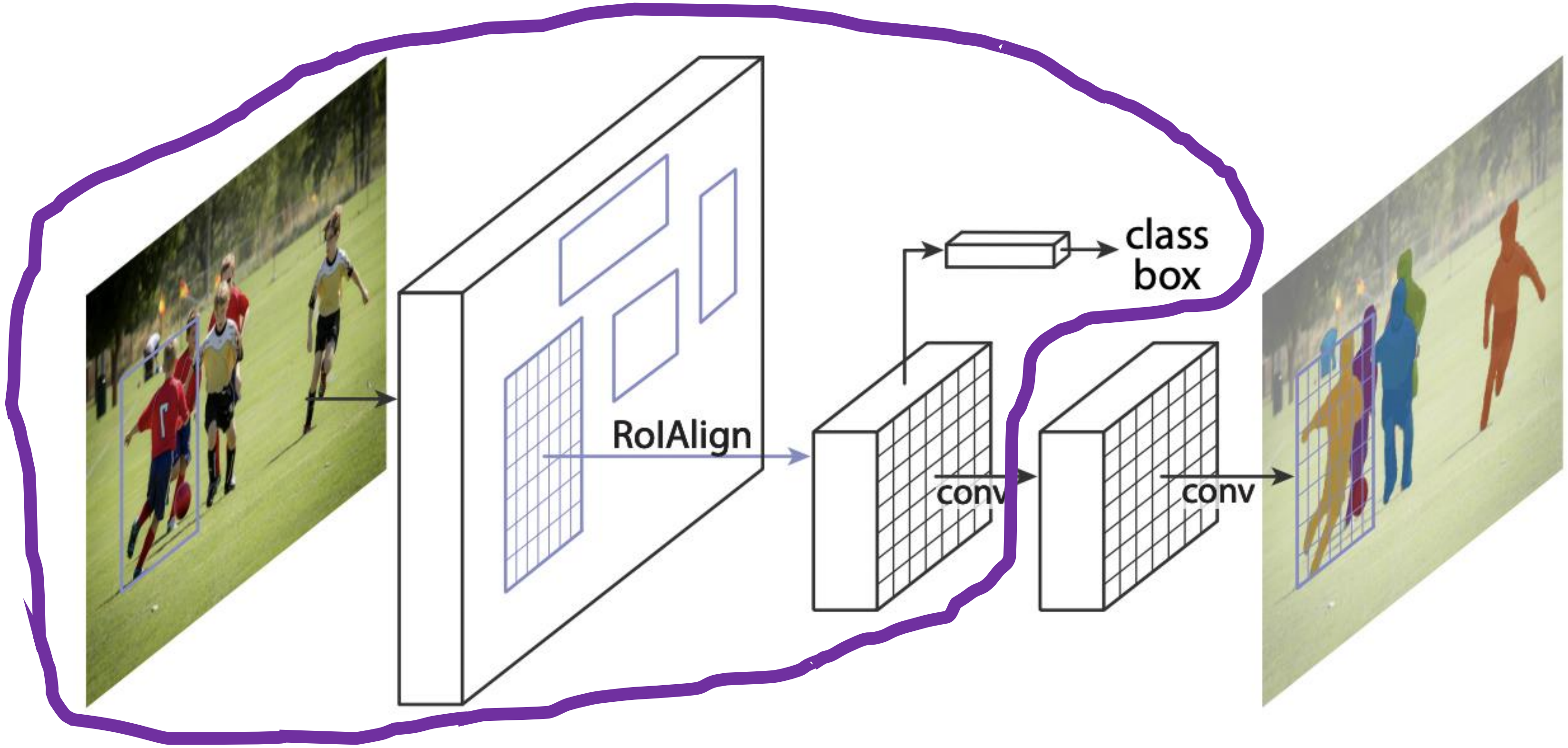
Instance Segmentation: Today's Topics

- Motivation
- Datasets
- Evaluation metric
- Mask R-CNN
- YOLACT

Historical Context



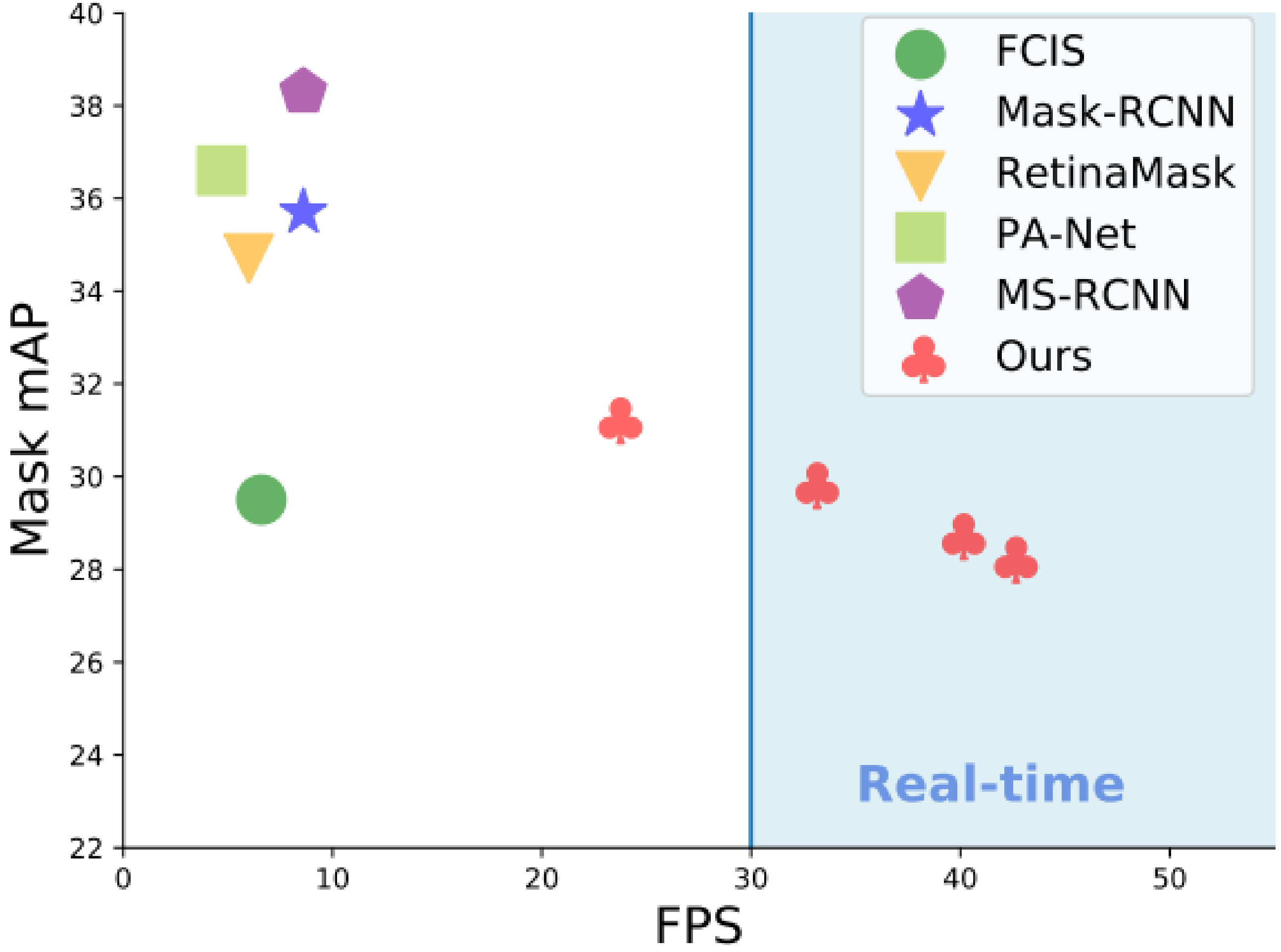
Motivation: Sequential 2-Stage Methods Are Slow



e.g., Faster R-CNN (1) generates features of a pre-defined size for each candidate region (i.e., output of the pooling method) which is then used for (2) mask prediction

He et al. Mask R-CNN. ICCV 2017

YOLACT Contribution: First Real-Time Instance Segmentation Model With Strong Performance



Bolya et al. YOLACT: Real-time Instance Segmentation. ICCV 2019

YOLACT Demo

https://www.youtube.com/watch?v=AJXCYks2_6s

Why YOLACT?

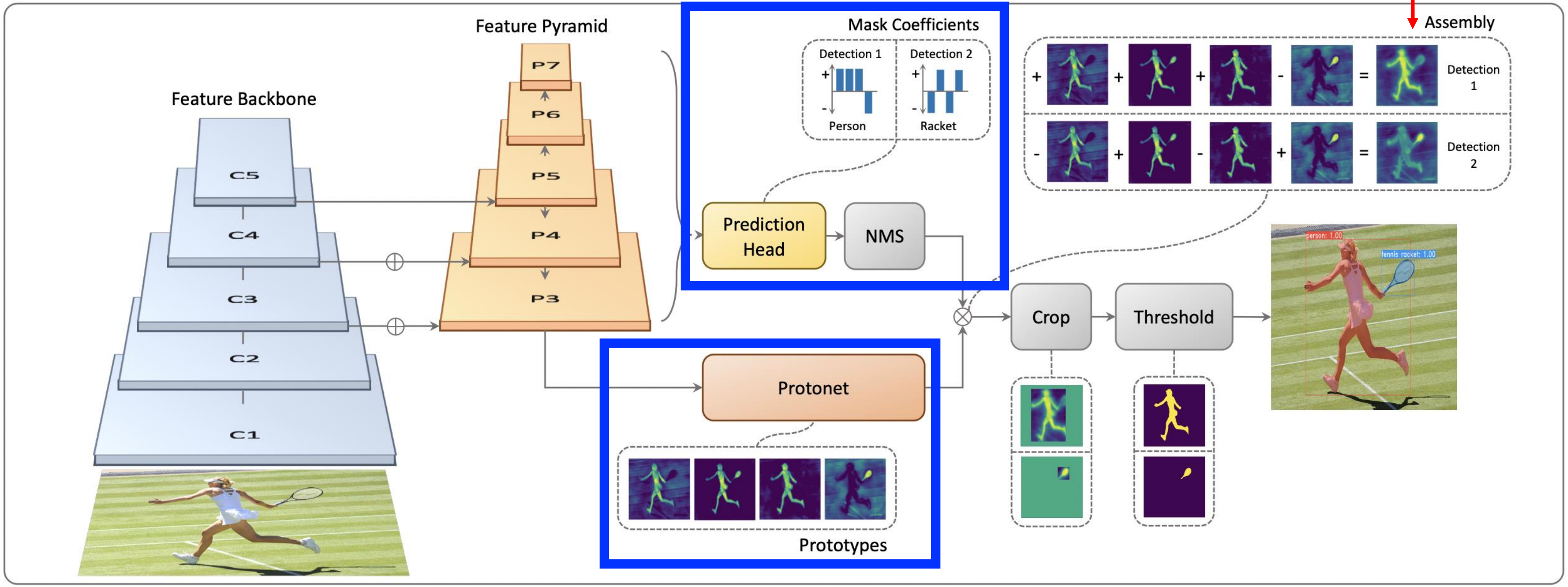
Named after the approach where **Y**ou **O**nly **L**ook **A**t **C**oefficien**t**s:

> Daniel Bolya, Chongy Zhou, Fanyi Xiao, & Yong Jae Lee. “YOLACT: Real-Time Instance Segmentation.” ICCV 2019.

Architecture: 1-Stage With Two **Parallel** Tasks (i.e., Doesn't Create Feature Per Region)

2. Predict per-instance mask coefficients

(Fast operation)



1. Generate *k* prototype masks (similar to semantic segmentation)

Bolya et al. YOLACT: Real-time Instance Segmentation. ICCV 2019

Training: Multi-Task Learning

- Matches Mask R-CNN with 3 losses for 3 tasks, while also augmenting a coefficient diversity loss

$$L = L_{class} + L_{box} + L_{mask}$$

Instance Segmentation: Today's Topics

- Motivation
- Datasets
- Evaluation metric
- Mask R-CNN
- YOLACT

The End