

## Motivation

### ❖ Prior Work:

- Scale Normalization for Image Pyramids:

- Idea: Only back-propagate gradients for objects within a size range when training on an image pyramid:



- ✗ Cons: Training on scales 1X, 2X, 3X is  $1 + 4 + 9 = 14X$  slower!

### ❖ Motivation

- Removing the redundant computation

- Is it possible to only process context regions around objects at higher resolutions of the image pyramid?



- Is context beyond a certain distance necessary?

## Challenges

- Skipped background regions may contain hard negative examples.
- Removing large portions of the background regions leads to higher False Positive rates

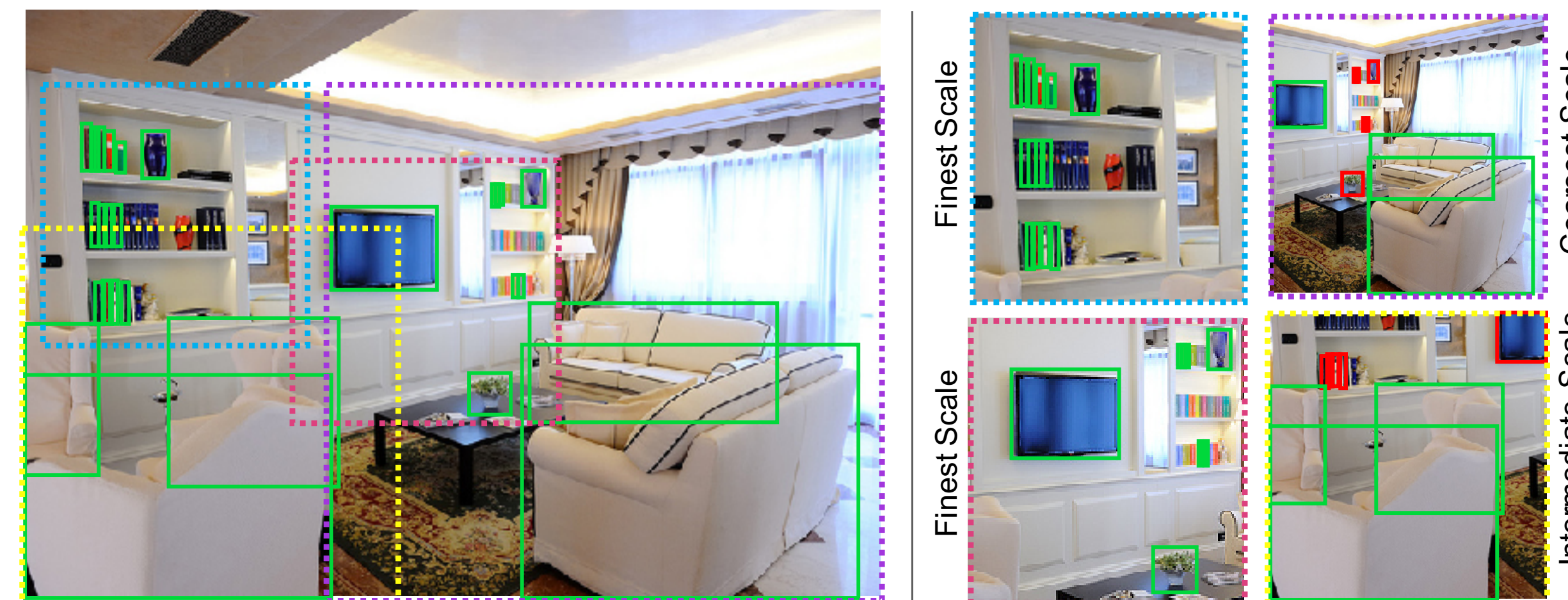
\* Authors contributed equally.

## SNIPER

- SNIPER operates on low-resolution positive and negative “chips”.

### ❖ Positive Chip Generation

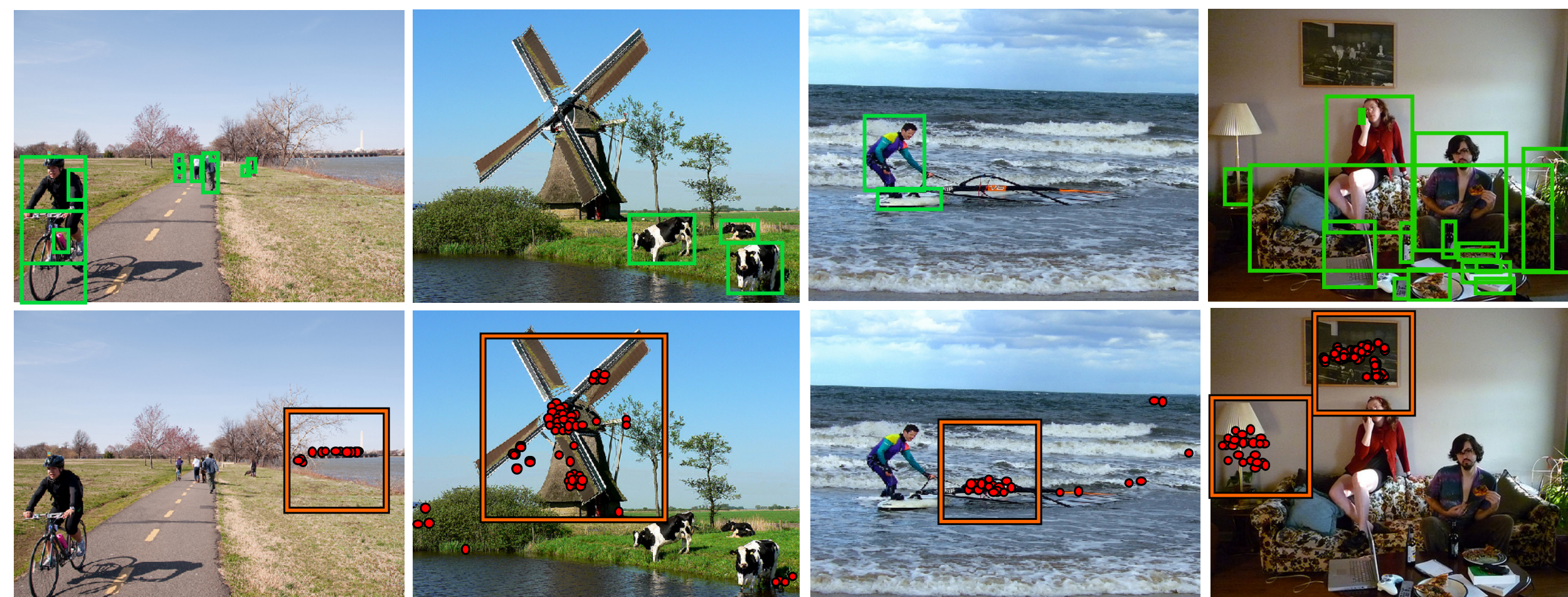
- Generate minimum number of chips while covering as many valid objects (“appropriate” for that scale) as possible.



Green boxes represent ground-truth objects. Frames with other colors represent chips.

### ❖ Negative Chip Generation

- Train RPN for a couple of epochs and generate chips covering RPN proposals not yet covered by positive chips.



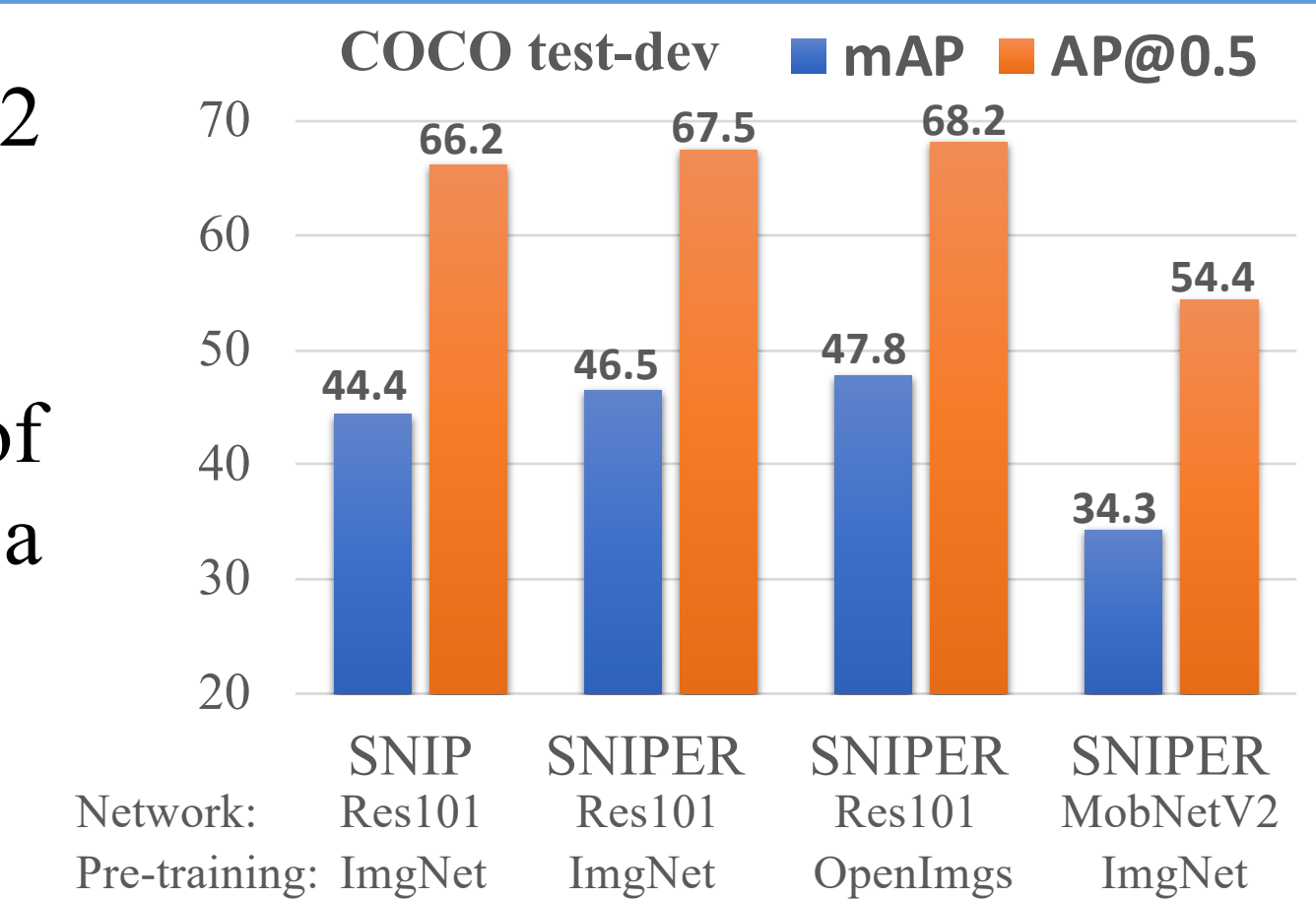
First row shows ground truth boxes. In the second row, proposals which are not covered in positive chips are shown as red dots. Red boxes are negative chips.

## Benefits

- ✓ Train Faster-RCNN with a batch size of 20 per GPU with a ResNet-101 backbone.
- ✓ Enables Batch-Normalization for instance-level recognition.
- ✓ No drop in performance compared to full resolution multi-scale training.
- ✓ Only 30% more pixels processed than  $800 \times 1333$  (the common single scale resolution for COCO).

## Results

- SNIPER is trained on  $512 \times 512$  chips with scales 1, 1.667 and 3.
- SNIPER achieves an mAP of 47.8% on COCO test-dev with a ResNet-101 backbone.



## “AutoFocus” & Code

- ❖ “[AutoFocus: Efficient Multi-Scale Inference](#)” is now on arXiv:
  - Efficient inference for SNIPER.
  - Chip generation is learned for inference.
  - More than 6 imgs/sec on a Titan X, same speed as RetinaNet with 10% higher mAP.
- ❖ SNIPER code is available on GitHub:
 

<http://github.com/mahyarnajibi/SNIPER>

