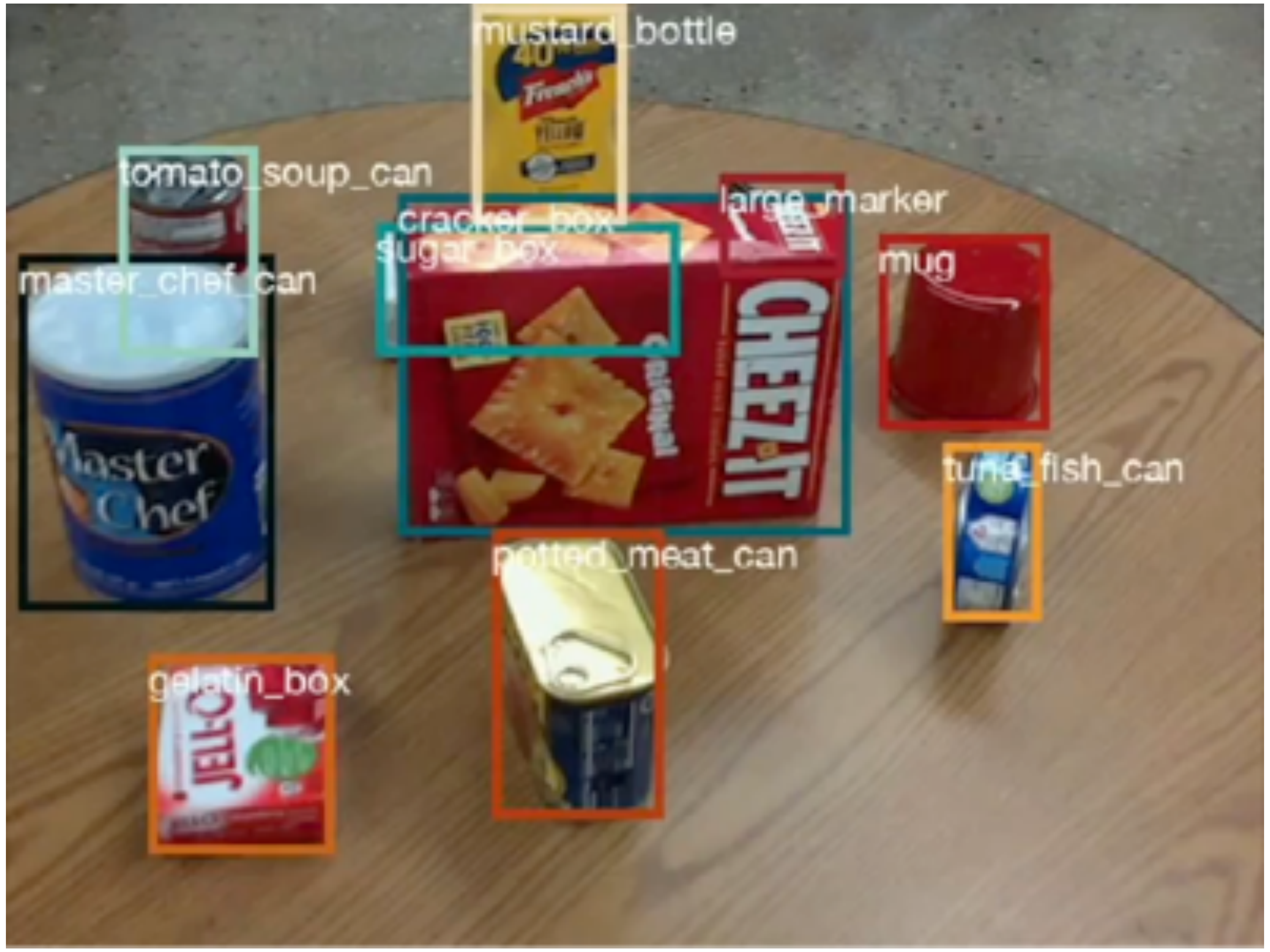
DeepRob

Lecture 13
Object Detectors and Segmentation
University of Michigan and University of Minnesota

Slides adapted from Justin Johnson's Deep Learning for Computer Vision course

Project 3 Updates

- Instructions available on the website
  - Here: https://rpm-lab.github.io/CSCI5980-Spr23-DeepRob/projects/project3/
  - New [PROPS Detection dataset]()
- Implement CNN for classification and Faster R-CNN for detection
- Autograder will be available soon!
- **Due Tuesday, March 14th 11:59 PM CT**



Final Project Tasks

1. [Graded] Final Project Proposal document submission (2%)
2. [Graded] In-class topic-paper(s) presentation (4%)
3. In-class final project pitch
4. In-class final project checkpoint
5. [Graded] Reproduce published results (12%)
    - Algorithmic extension to obtain results with new idea, technique or dataset
6. [Graded] Video Presentation + Poster (4%)
7. [Graded] Final Report (2%)

Final Project Tasks

1. [Graded] Final Project Proposal document submission (2%)
2. [Graded] In-class to ic-paper(s) presentation (4%)
3. In-class final project p h
4. In-class final project c
5. [Graded] Reproduce
   - Algorithmic extens
6. [Graded] Video Prese
7. [Graded] Final Repor

Paper selection **done!**

Final Project Proposal due 03/02

- LaTeX template was shared over canvas announcement
- Email the proposals to me. See details in the announcement

I will give feedback next week that you can use for the project pitches!

Final Project Tasks

1. [Graded] Final Project Proposal document submission (2%)
2. [Graded] In-class topic-paper(s) presentation (4%)
3. In-class final project itch
4. In-class final project c kpoint
5. [Graded] Reproduce
   - Algorithmic extens set
6. [Graded] Video Prese
7. [Graded] Final Repor

Student lecture-presentations starting next lecture

If you presenting on a Tuesday

Meet with me during OH the previous Wednesday

If you presenting on a Thursday

Meet with me during OH the previous Friday

Final Project Tasks

1. [Graded] Final Project Proposal document submission (2%)
2. [Graded] In-class topic-paper(s) presentation (4%)
3. In-class final project pitch
4. In-class final proje
5. [Graded] Reproduce
    - Algorithmic exter
6. [Graded] Video Pres
7. [Graded] Final Repo

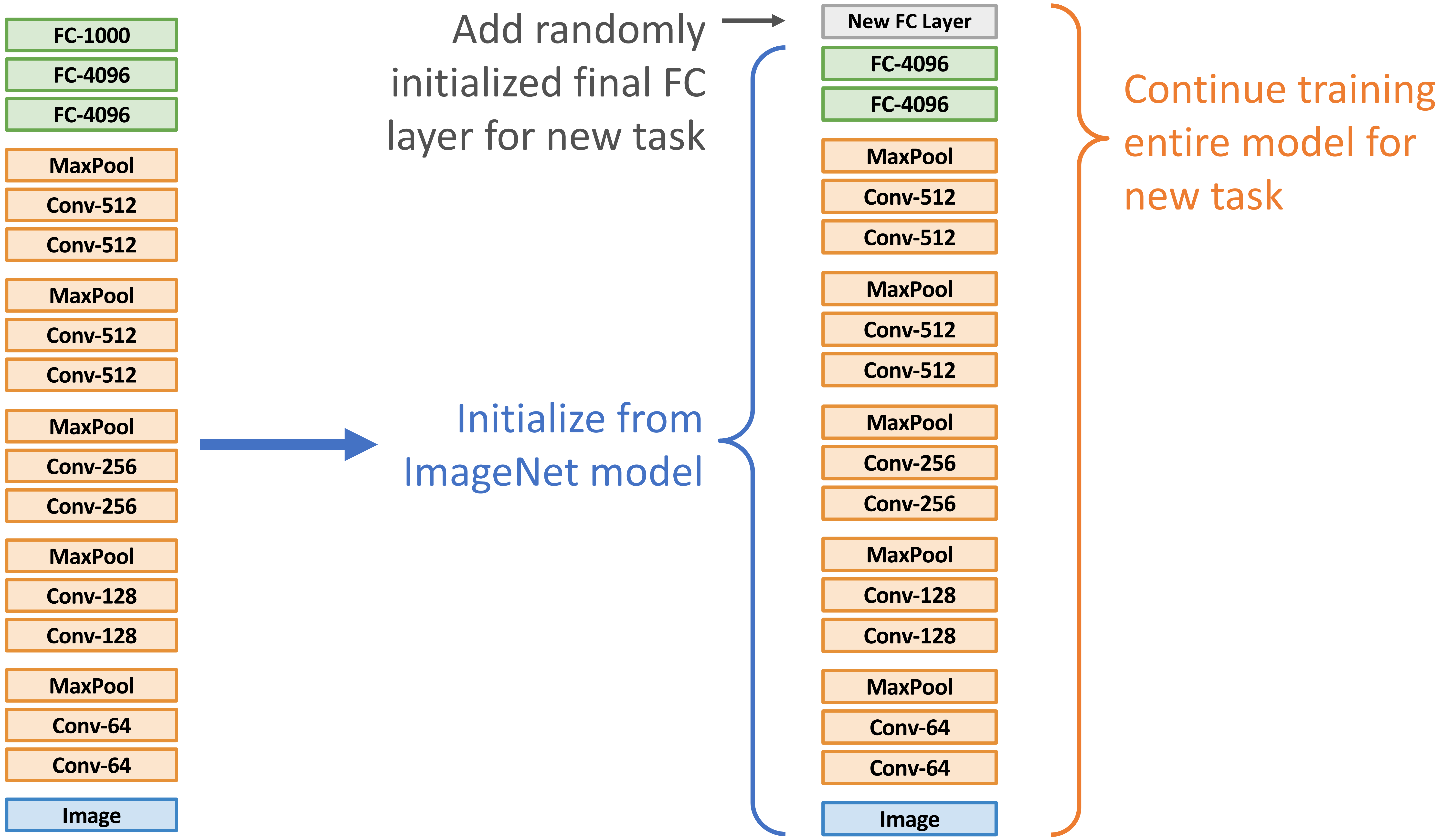
In-class 03/14!

Each team is required to a upload a recorded video (max 8min length) before 2pm on 03/14.

Others will rate the project pitch
- Compelling Pitch?
- Is the effort distribution among members clear?
- Is the effort distribution even?
- Plan A and Plan B clear?

Last time: Transfer Learning

1. Train on ImageNet



Add randomly initialized final FC layer for new task

Initialize from ImageNet model

Continue training entire model for new task

Last time: Localization Tasks

Classification

"Chocolate Pretzels"

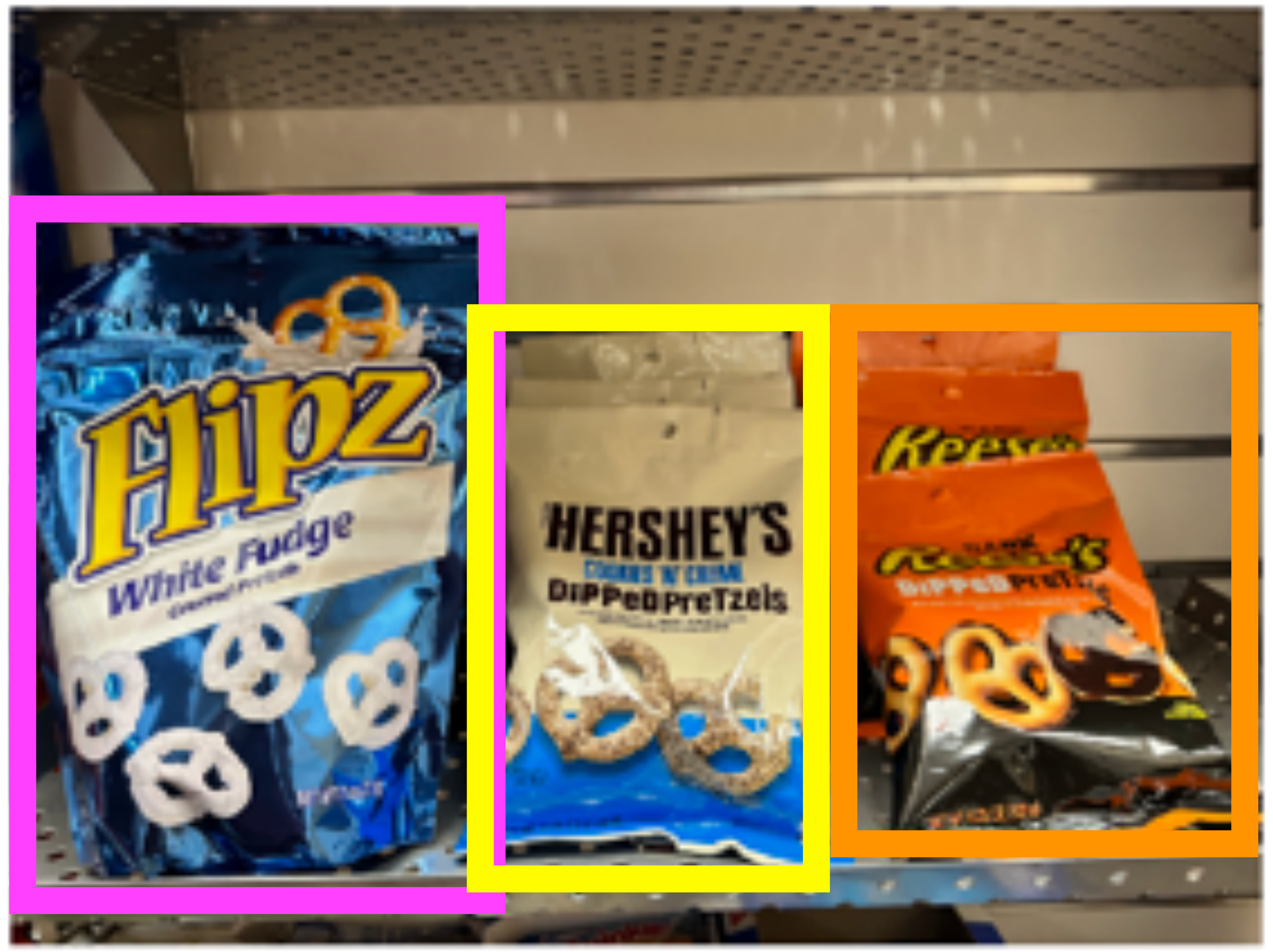
No spatial extent

Semantic Segmentation

Chocolate Pretzels, Shelf

No objects, just pixels

Object Detection



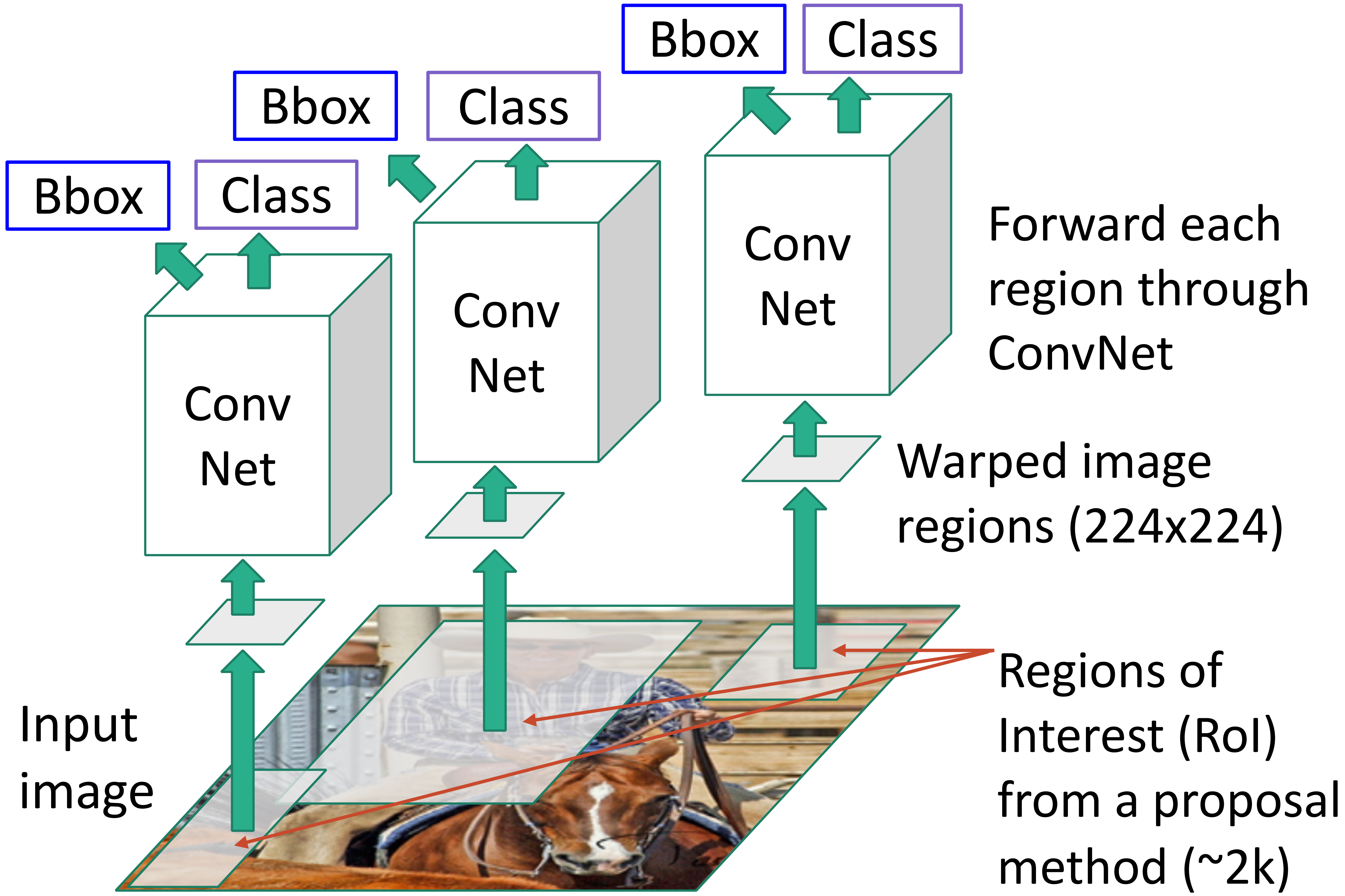
Instance Segmentation

Flipz, Hershey's, Keese's

Multiple objects

Last time: R-CNN

R-CNN: Region-Based CNN

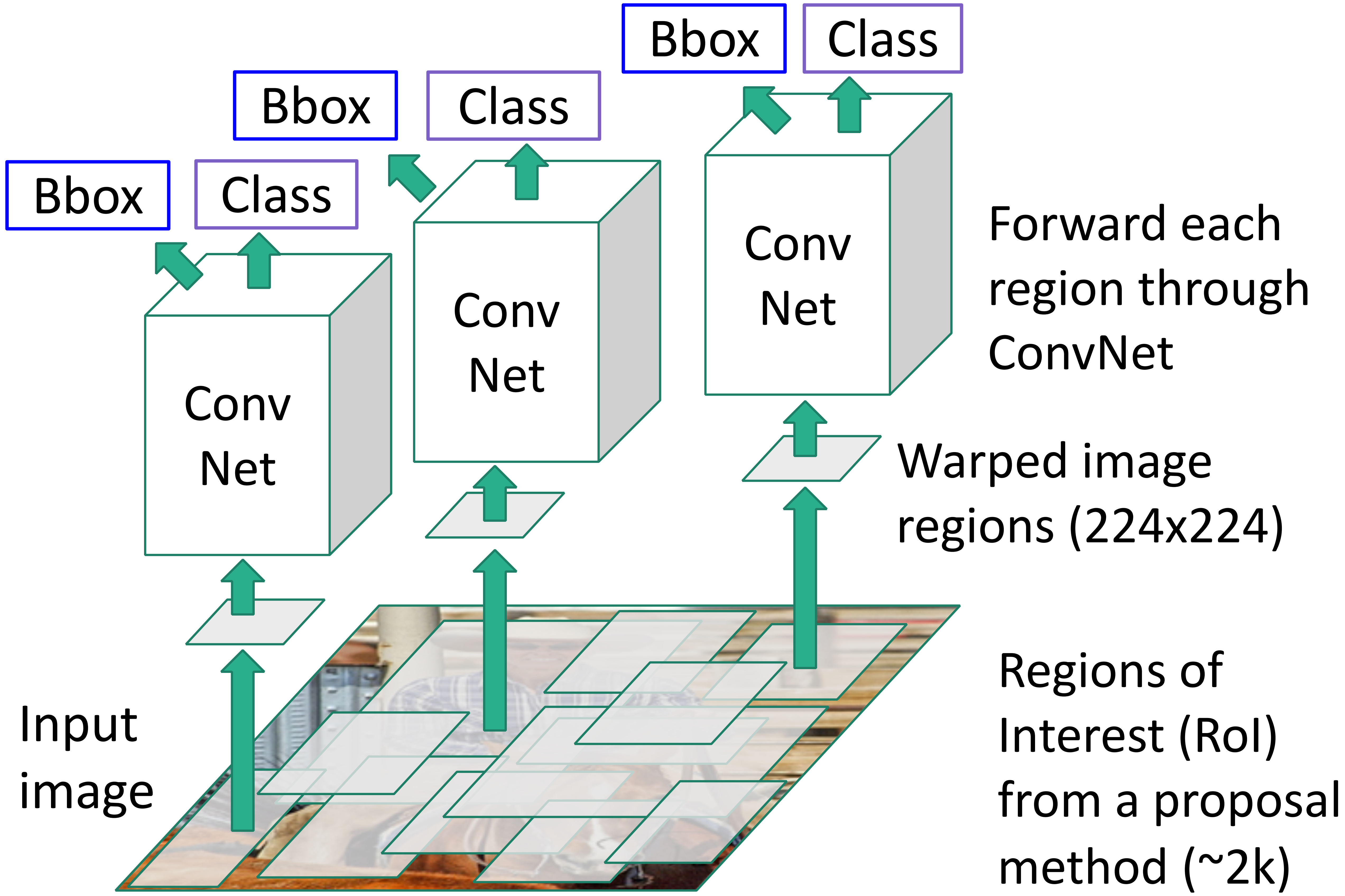


Classify each region

Bounding box regression:
Predict "transform" to correct the RoI: 4 numbers (t_x, t_y, t_h, t_w)

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission

Last time: R-CNN



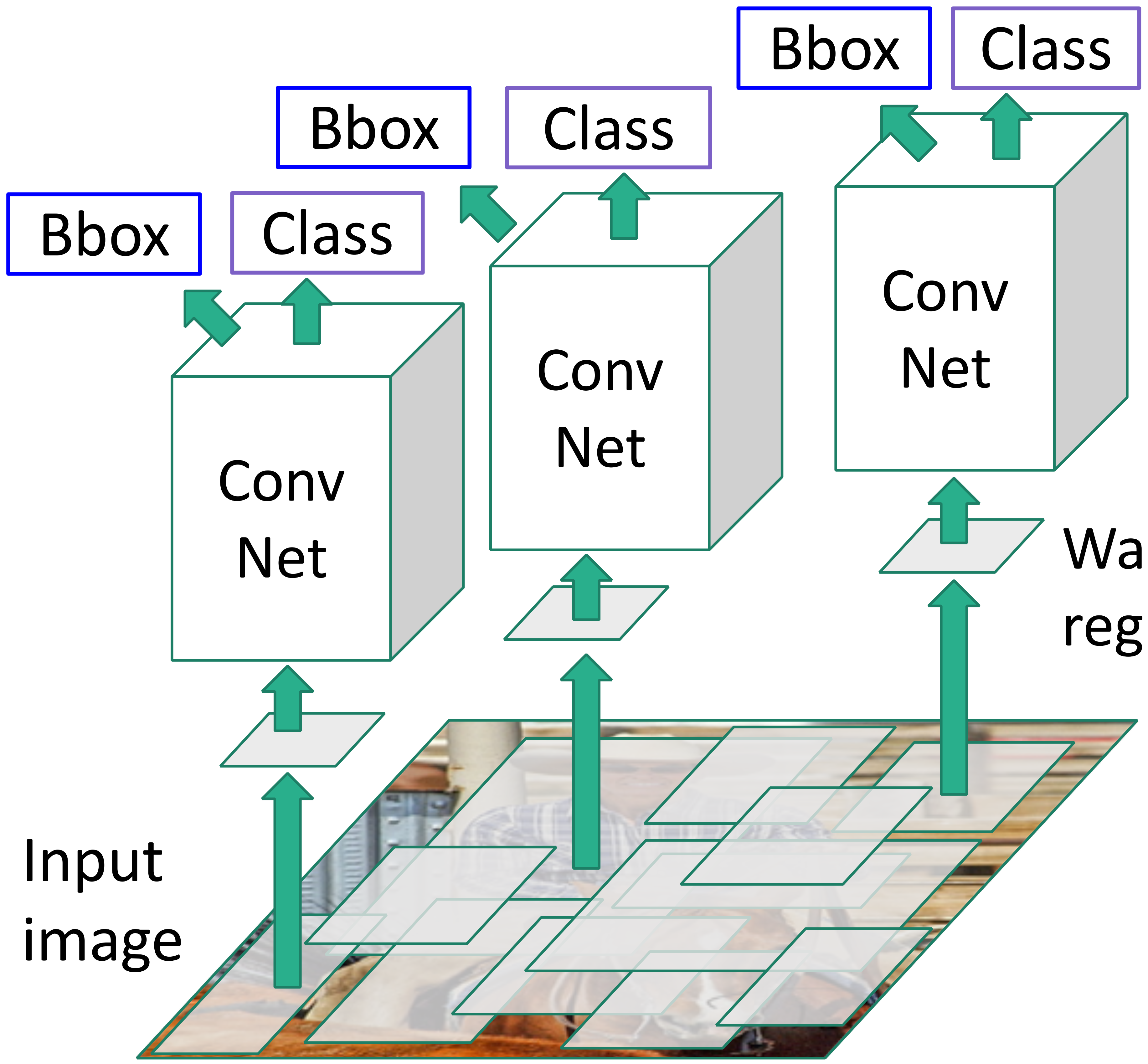
Classify each region

Bounding box regression:
Predict "transform" to correct the RoI: 4 numbers (t_x, t_y, t_h, t_w)

Problem: Very slow! Need to do 2000 forward passes through CNN per image

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission

Last time: R-CNN



Forward each region through ConvNet

Warped image regions (224x224)

Regions of Interest (RoI) from a proposal method (~2k)

Input image

Classify each region

Bounding box regression: Predict "transform" to correct the RoI: 4 numbers (t_x, t_y, t_h, t_w)

Problem: Very slow! Need to do 2000 forward passes through CNN per image

Idea: Overlapping proposals cause a lot of repeated work; same pixels processed many times. Can we avoid this?

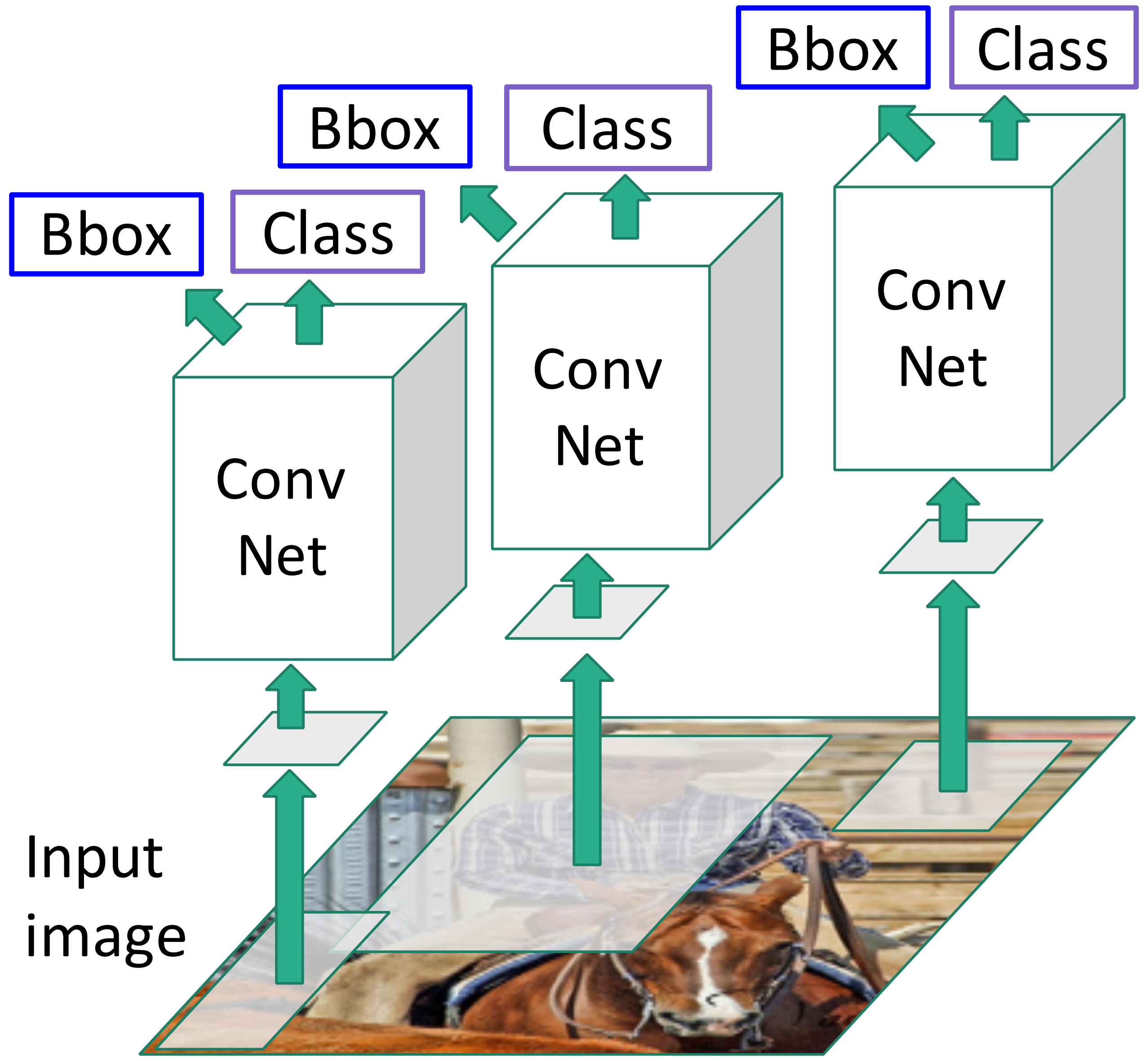
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission

Fast R-CNN


Input image

<u>“Slow” R-CNN</u>

Process each region independently



Fast R-CNN

“Backbone” network:
AlexNet, VGG,
ResNet, etc

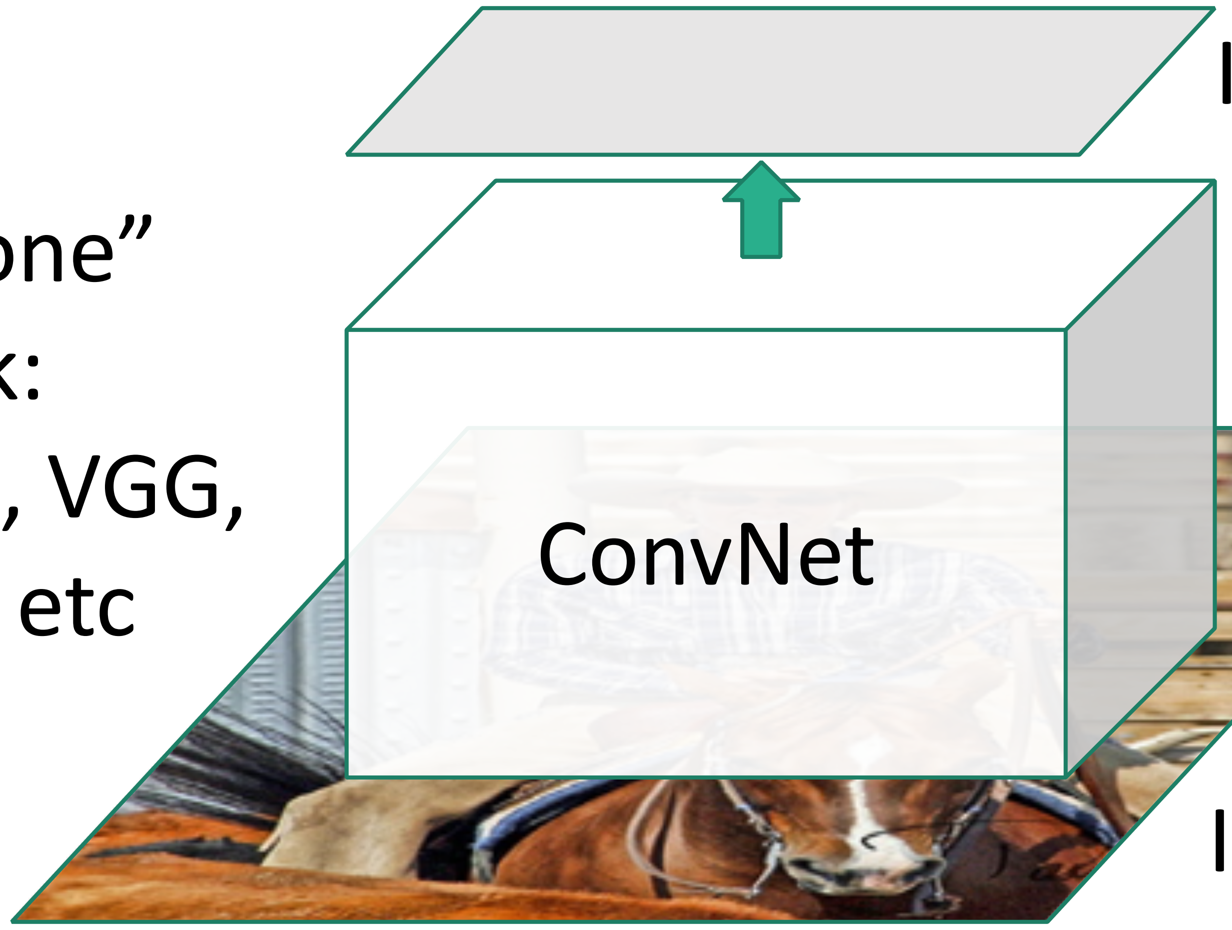


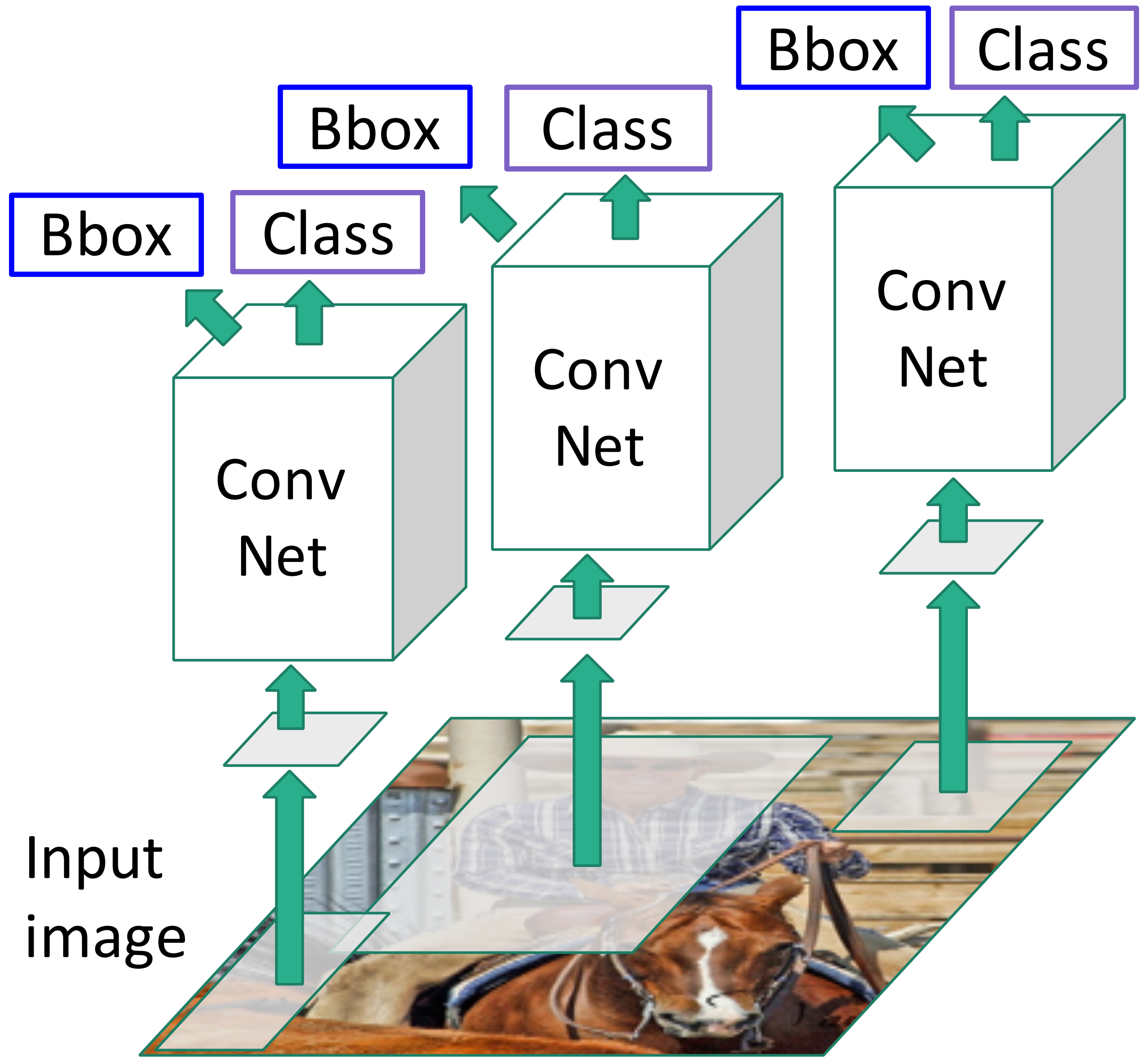
Image features

Run whole image through ConvNet

ConvNet

Input image

<u>“Slow” R-CNN</u>
Process each region independently



Fast R-CNN

Regions of Interest (RoIs) from a proposal method

"Backbone" network: AlexNet, VGG, ResNet, etc

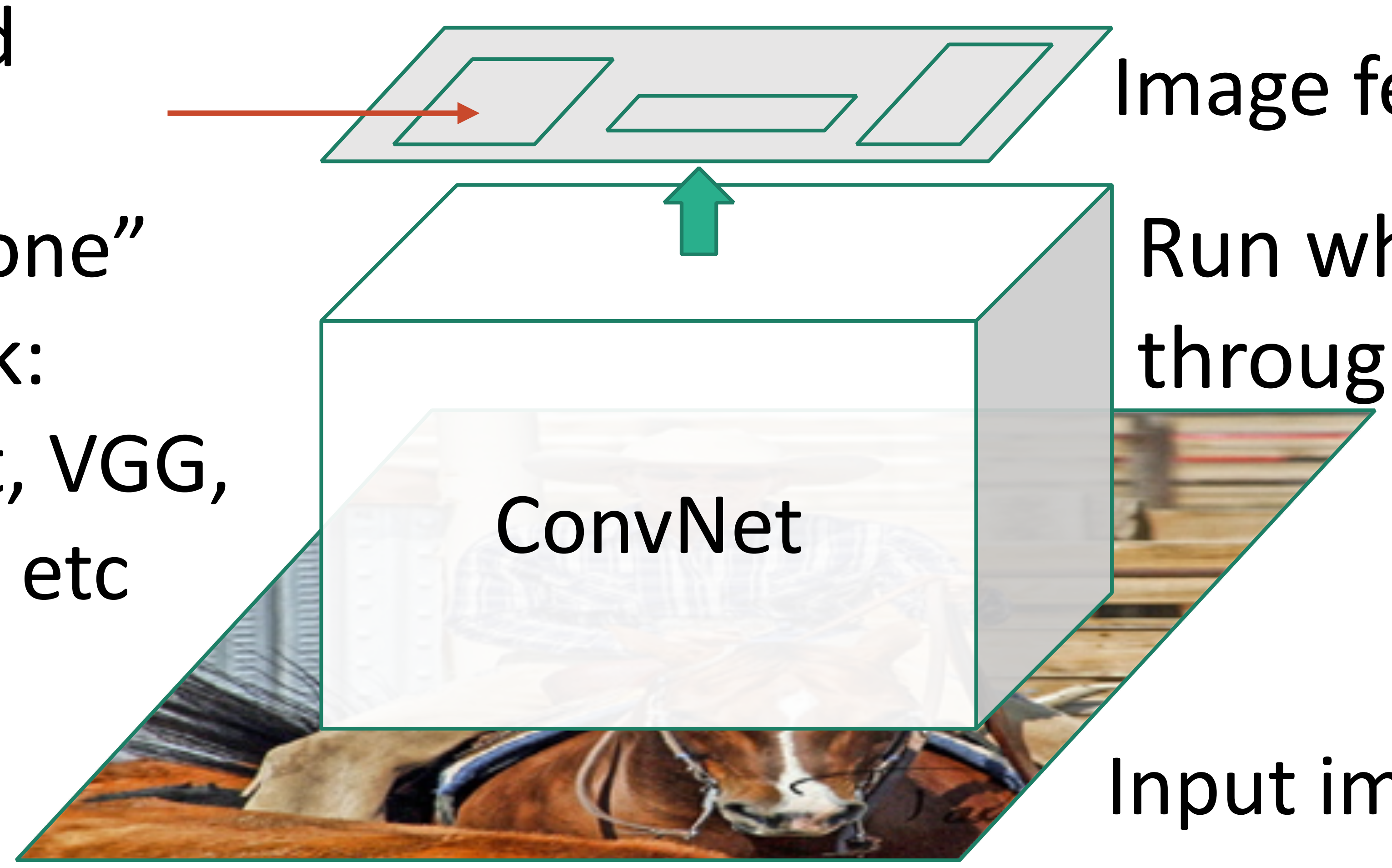


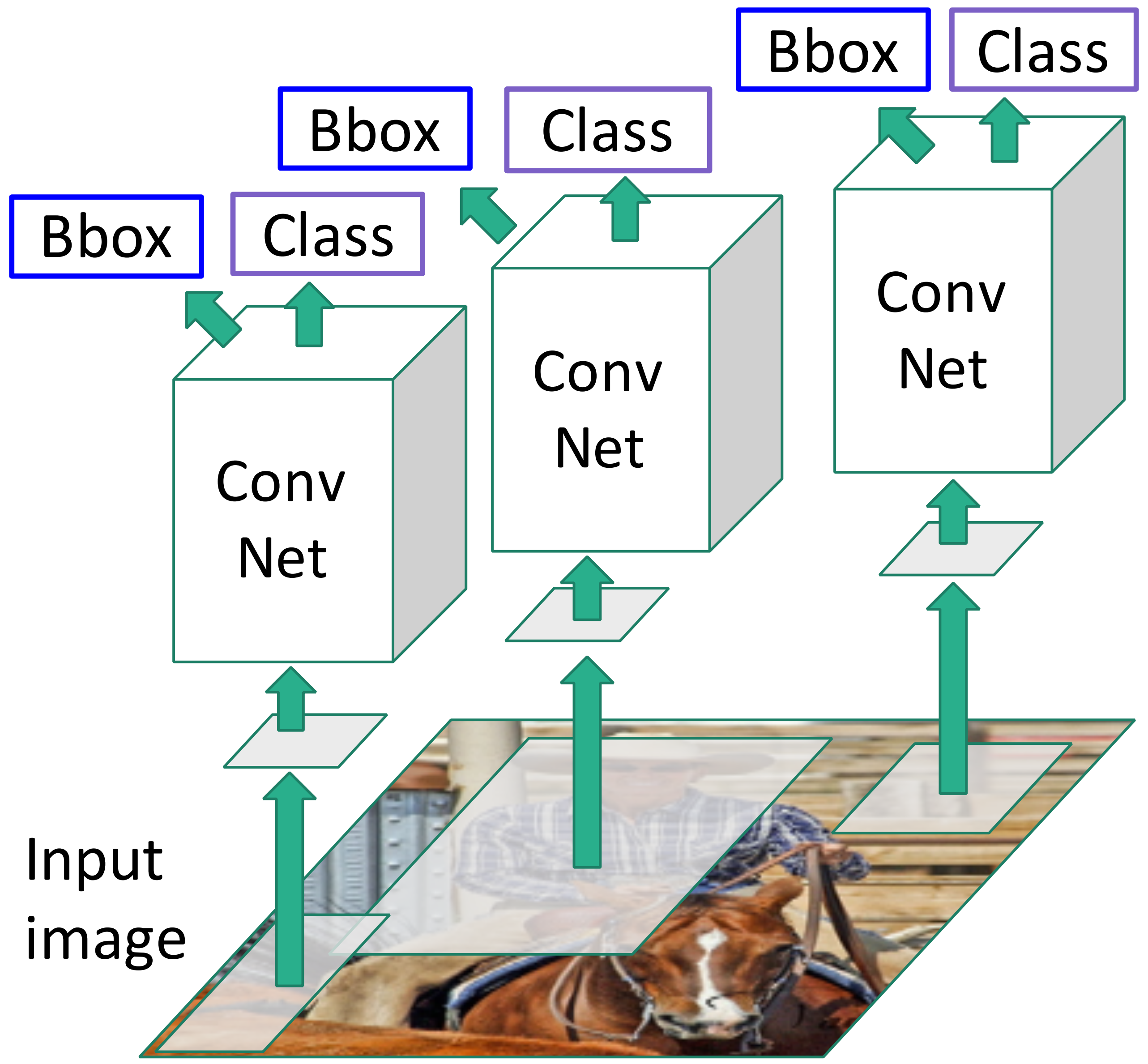
Image features

Run whole image through ConvNet

Input image

<u>"Slow" R-CNN</u>

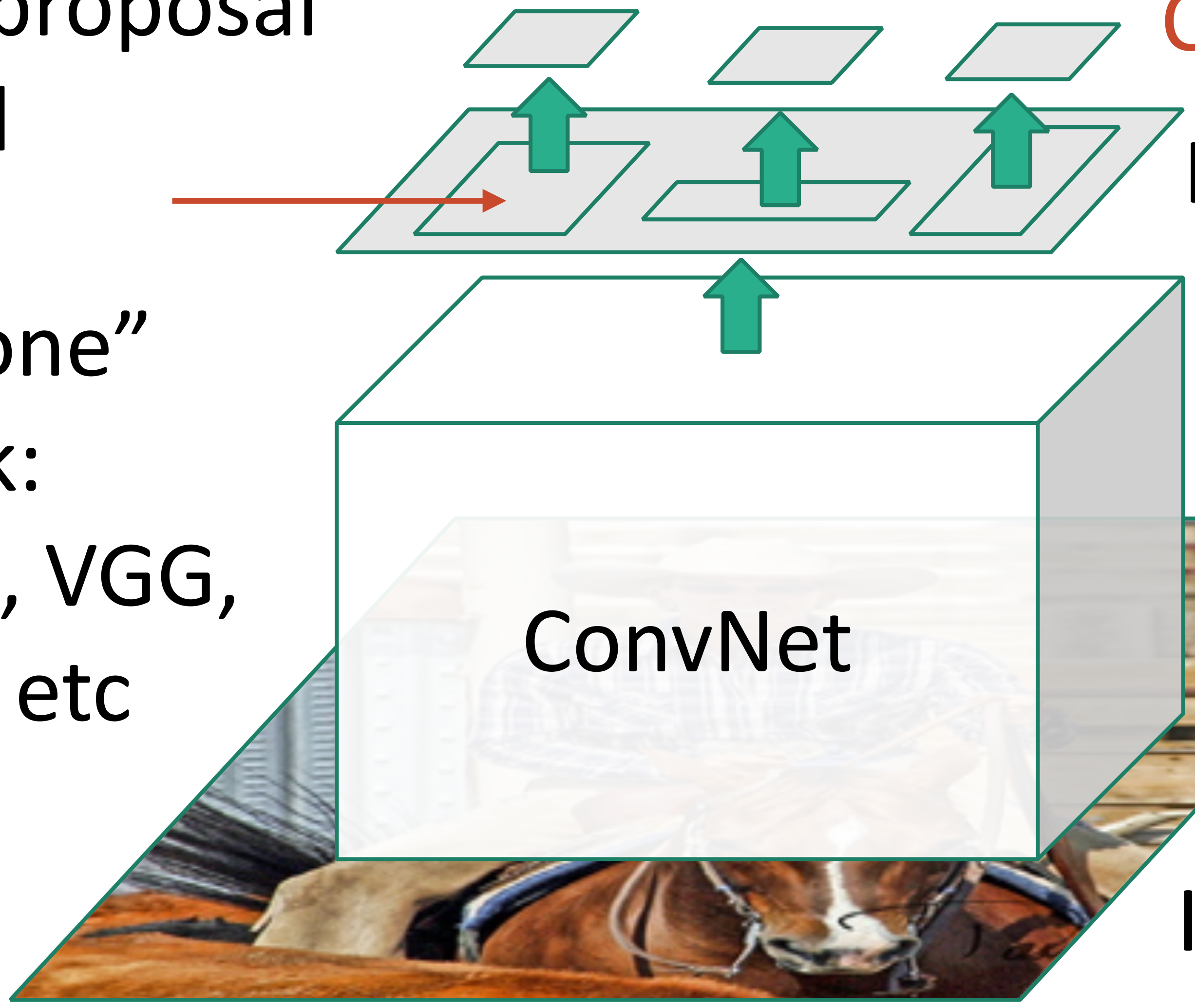
Process each region independently



Fast R-CNN

Regions of Interest (RoIs) from a proposal method

“Backbone” network: AlexNet, VGG, ResNet, etc



Crop + Resize features

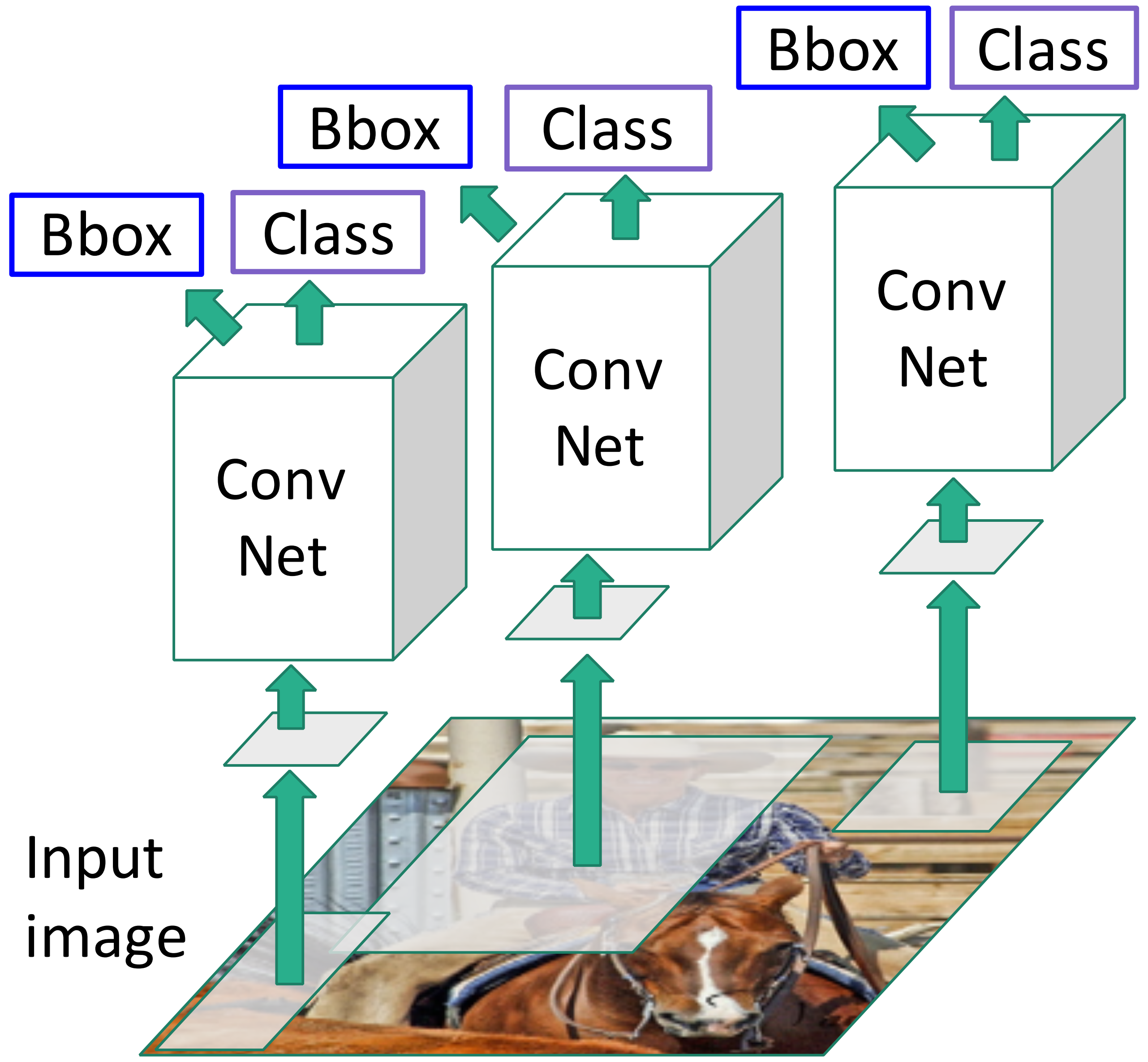
Image features

Run whole image through ConvNet

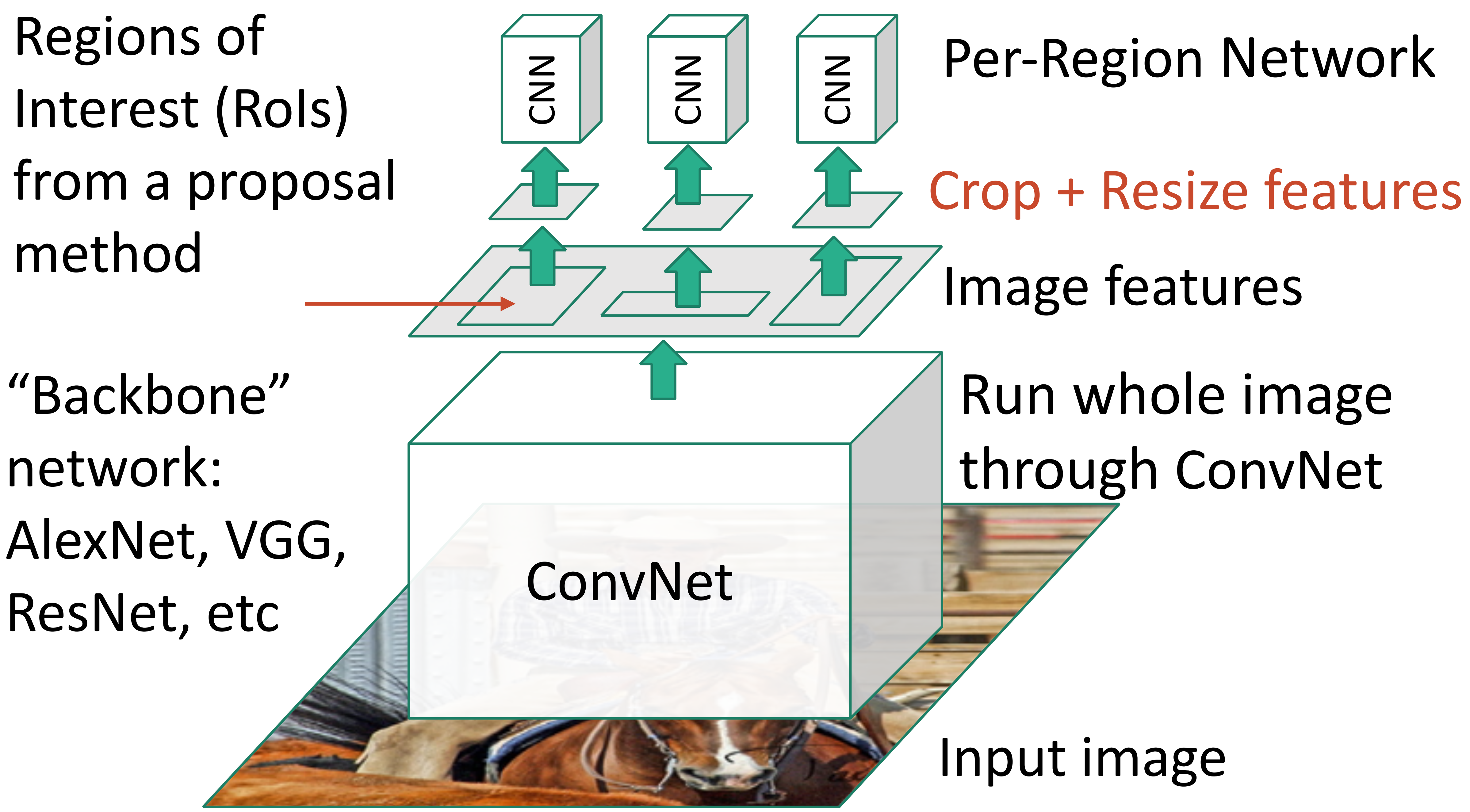
Input image

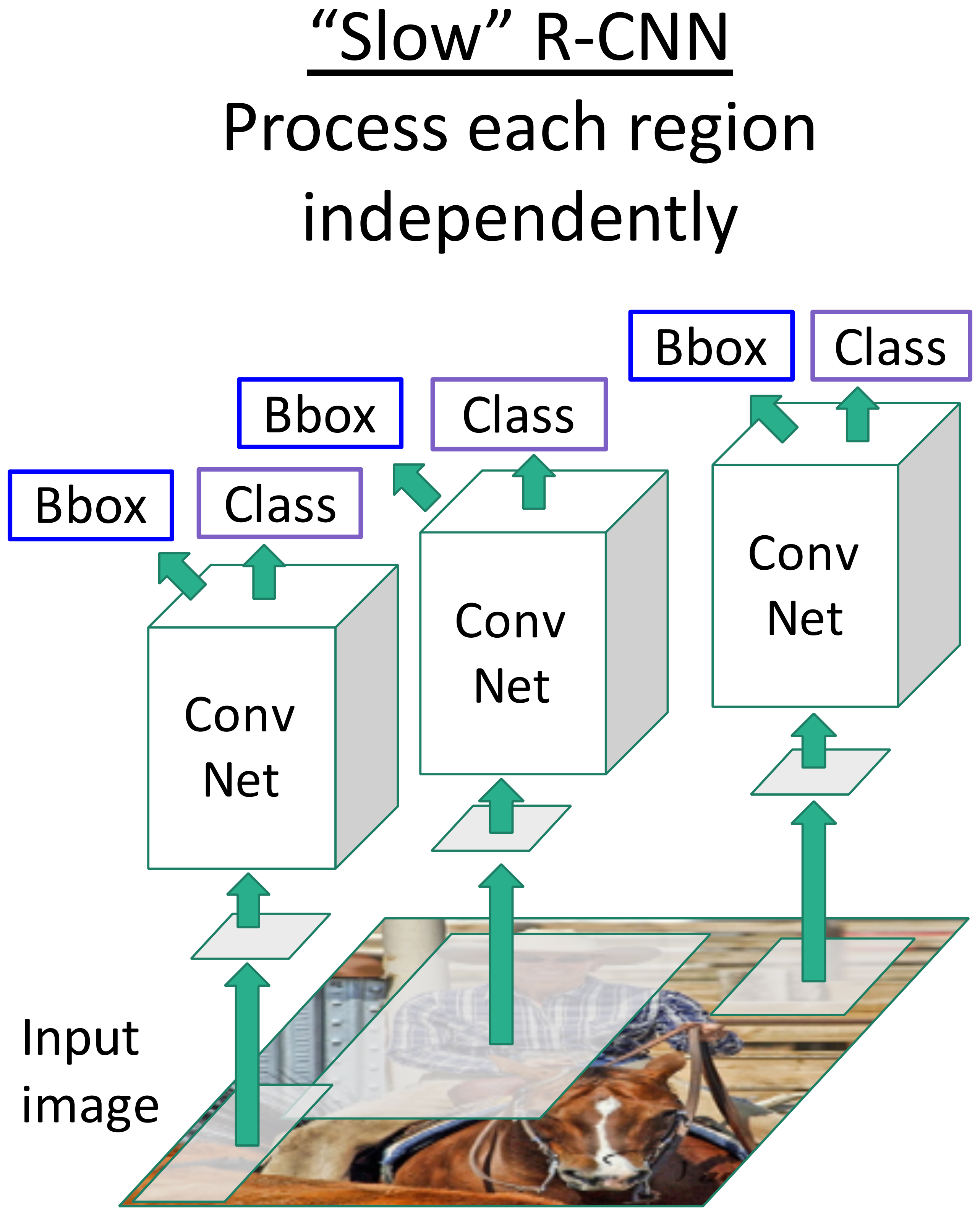
“Slow” R-CNN

Process each region independently

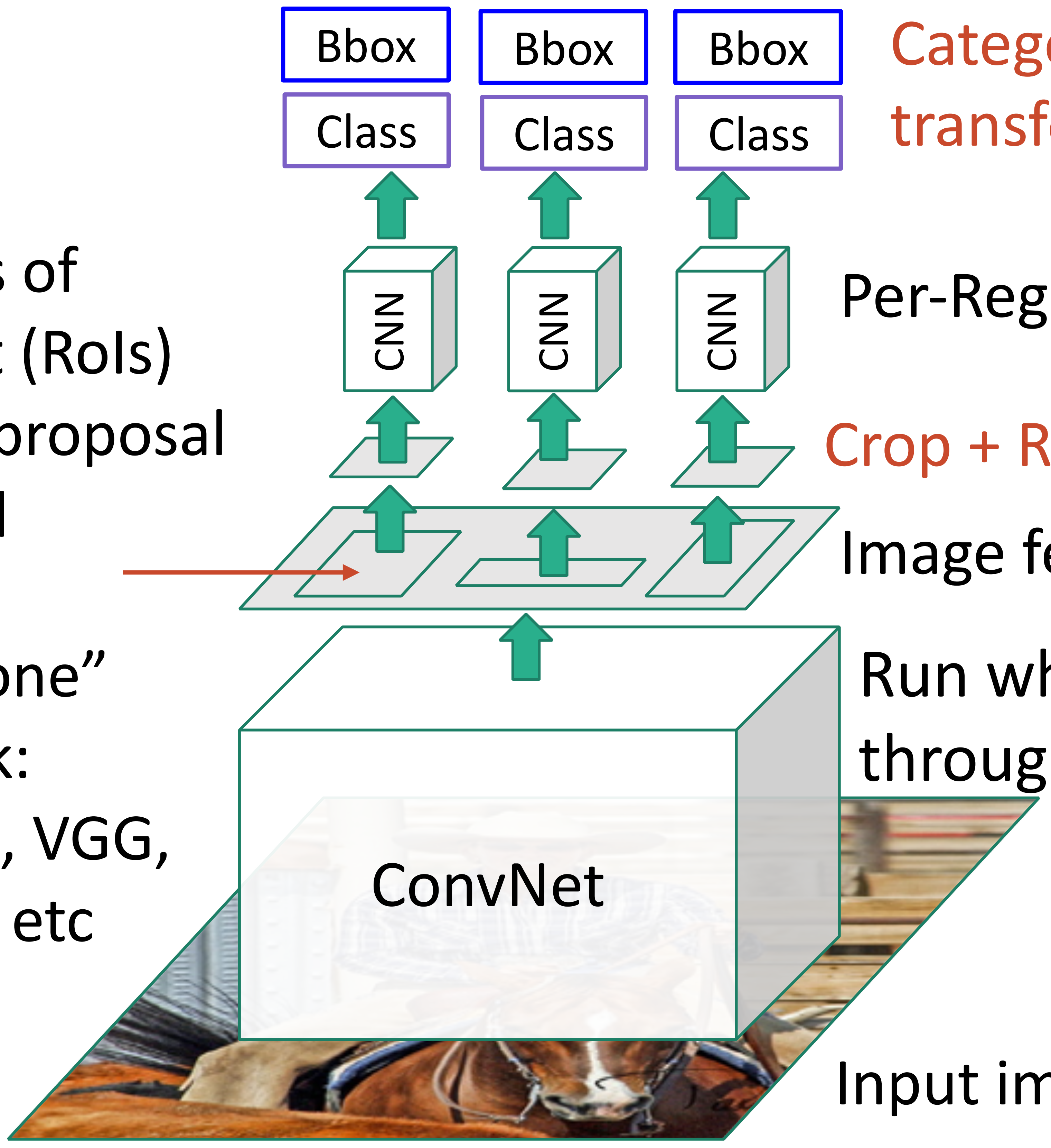


Fast R-CNN


Regions of Interest (RoIs) from a proposal method
CNN
CNN
CNN
Per-Region Network
Crop + Resize features
Image features
"Backbone" network: AlexNet, VGG, ResNet, etc
Run whole image through ConvNet
ConvNet
Input image


"Slow" R-CNN
Process each region independently
Bbox
Class
Bbox
Class
Bbox
Class
Conv Net
Conv Net
Conv Net
Input image

Fast R-CNN



Category and box transform per region

Per-Region Network

Regions of Interest (RoIs) from a proposal method

Crop + Resize features

Image features

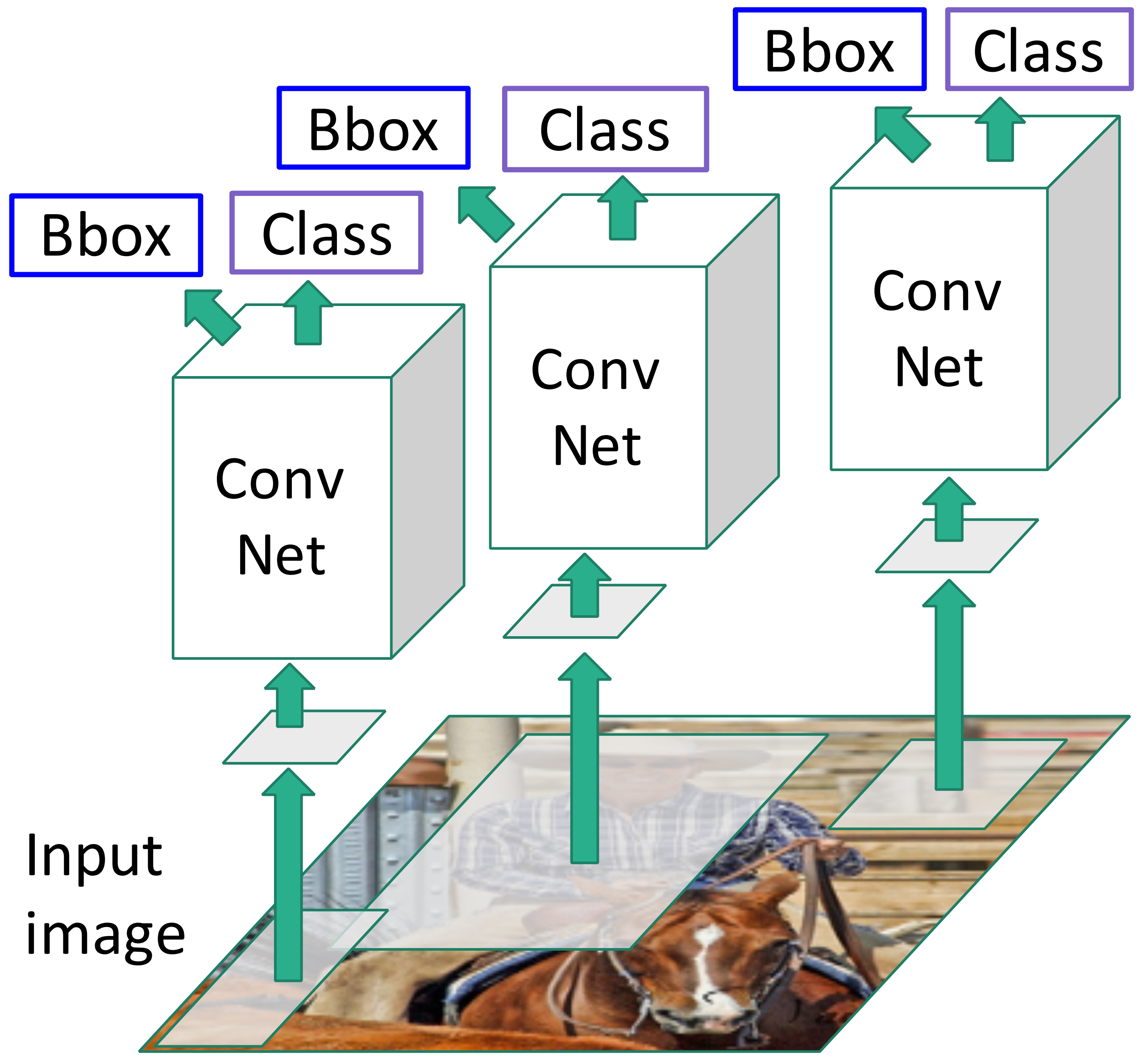
"Backbone" network: AlexNet, VGG, ResNet, etc

Run whole image through ConvNet

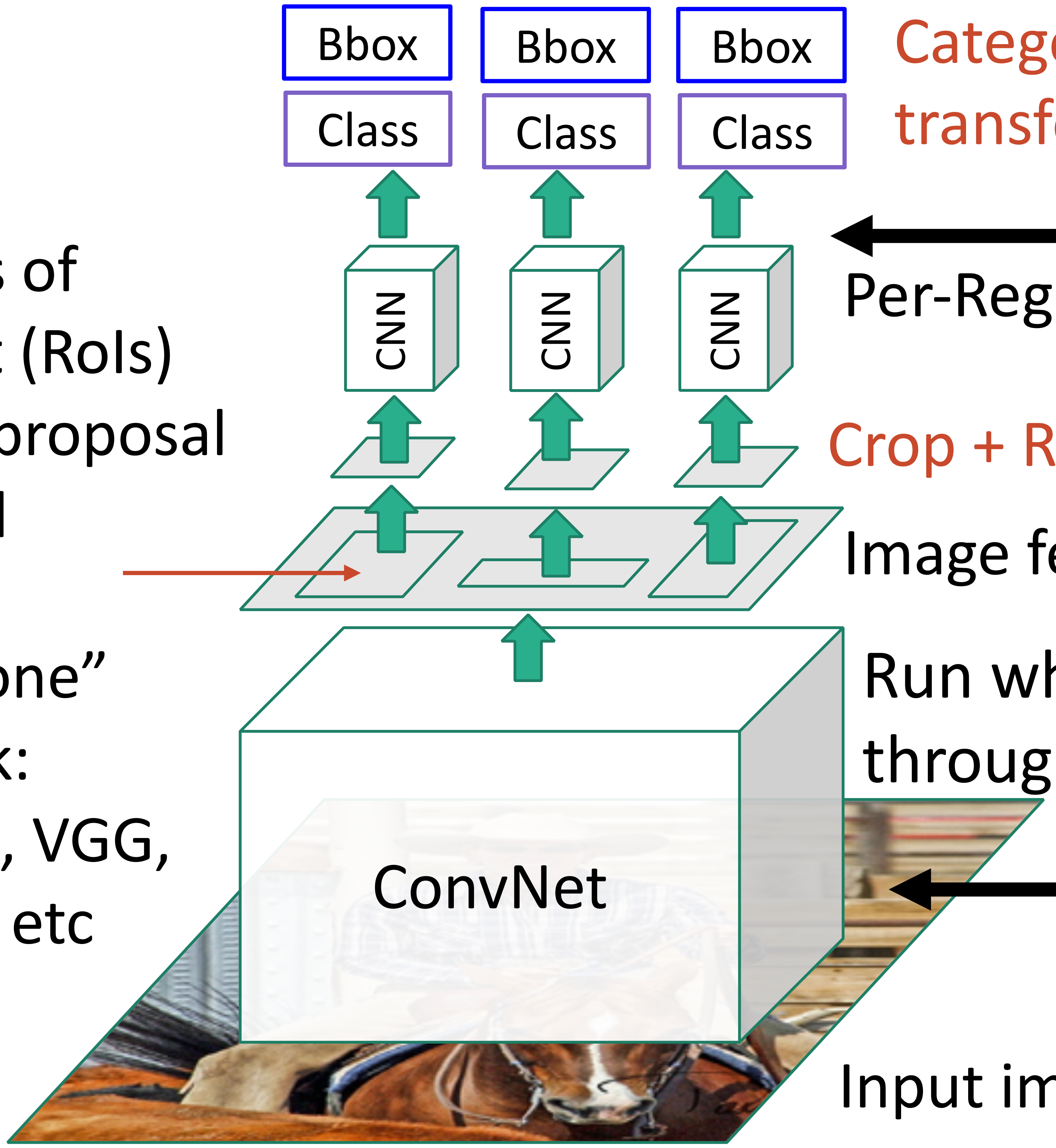
Input image

"Slow" R-CNN

Process each region independently



Fast R-CNN



Category and box transform per region

Per-Region network is relatively lightweight

Per-Region Network

Regions of Interest (RoIs) from a proposal method

Crop + Resize features

Image features

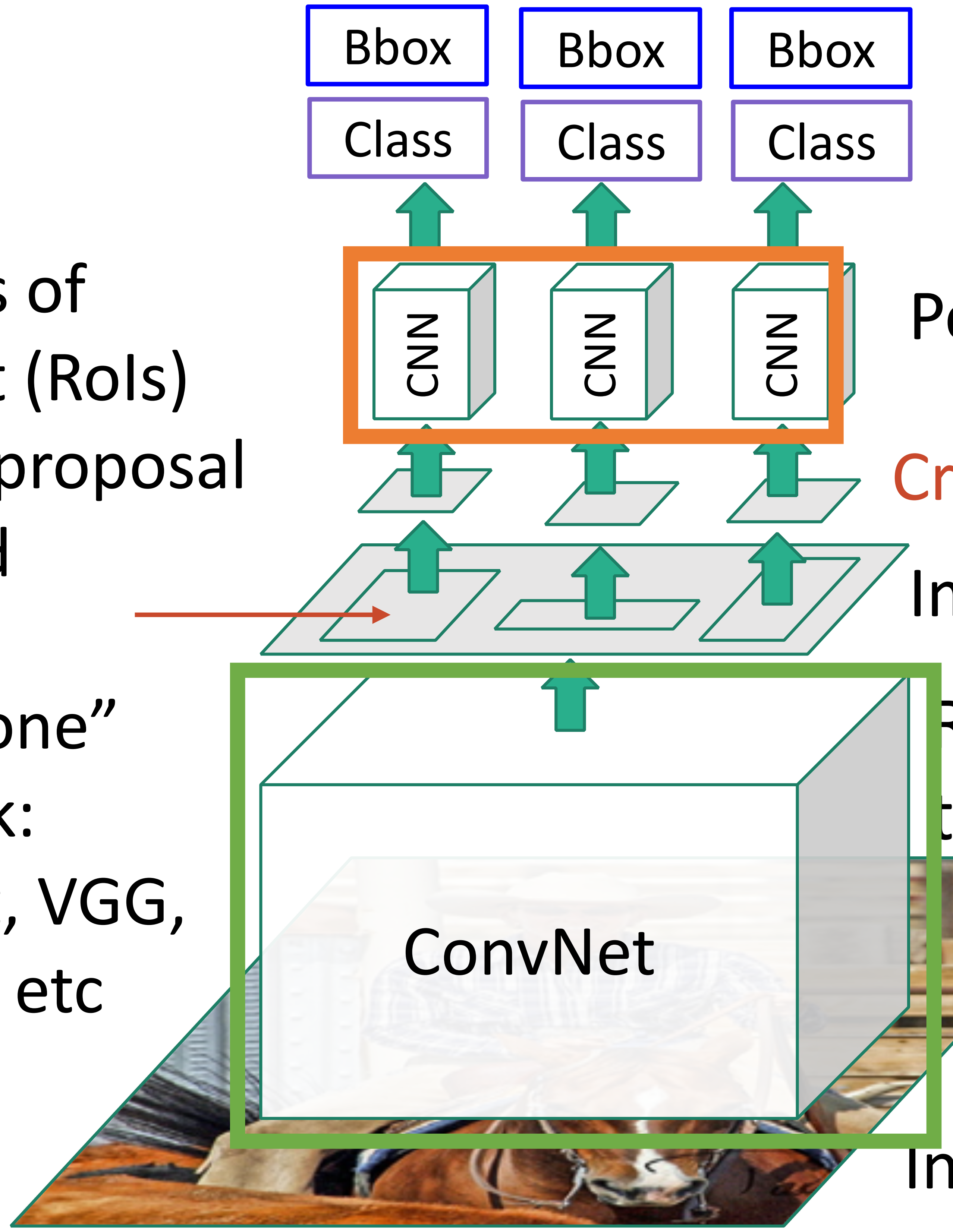
"Backbone" network: AlexNet, VGG, ResNet, etc

Run whole image through ConvNet

Most of the computation happens in backbone network; this saves work for overlapping region proposals

Input image

Fast R-CNN



Category and box transform per region

Per-Region Network

Crop + Resize features

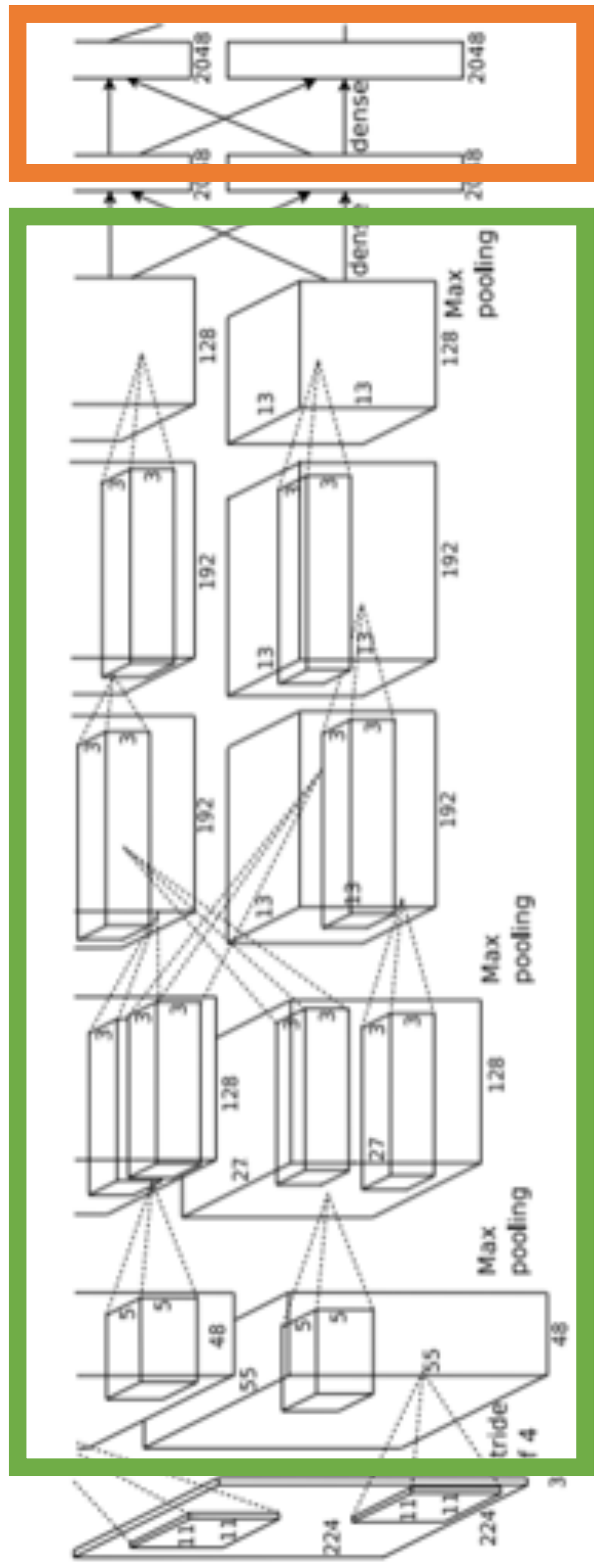
Image features

Regions of Interest (RoIs) from a proposal method

"Backbone" network: AlexNet, VGG, ResNet, etc

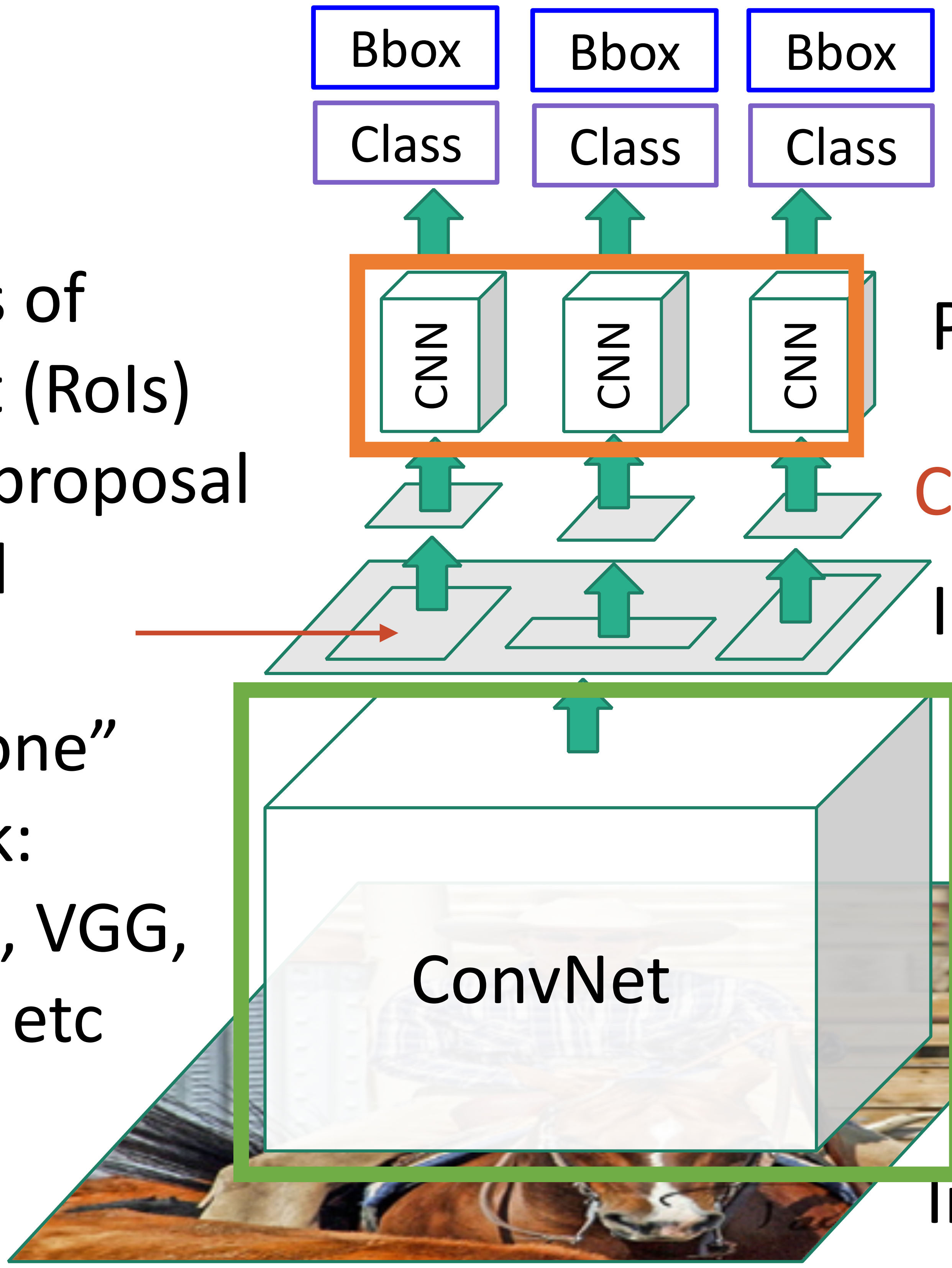
Run whole image through ConvNet

Input image



Example: When using AlexNet for detection, five conv layers are used for backbone and two FC layers are used for per-region network

Fast R-CNN



Category and box transform per region

Per-Region Network

Crop + Resize features

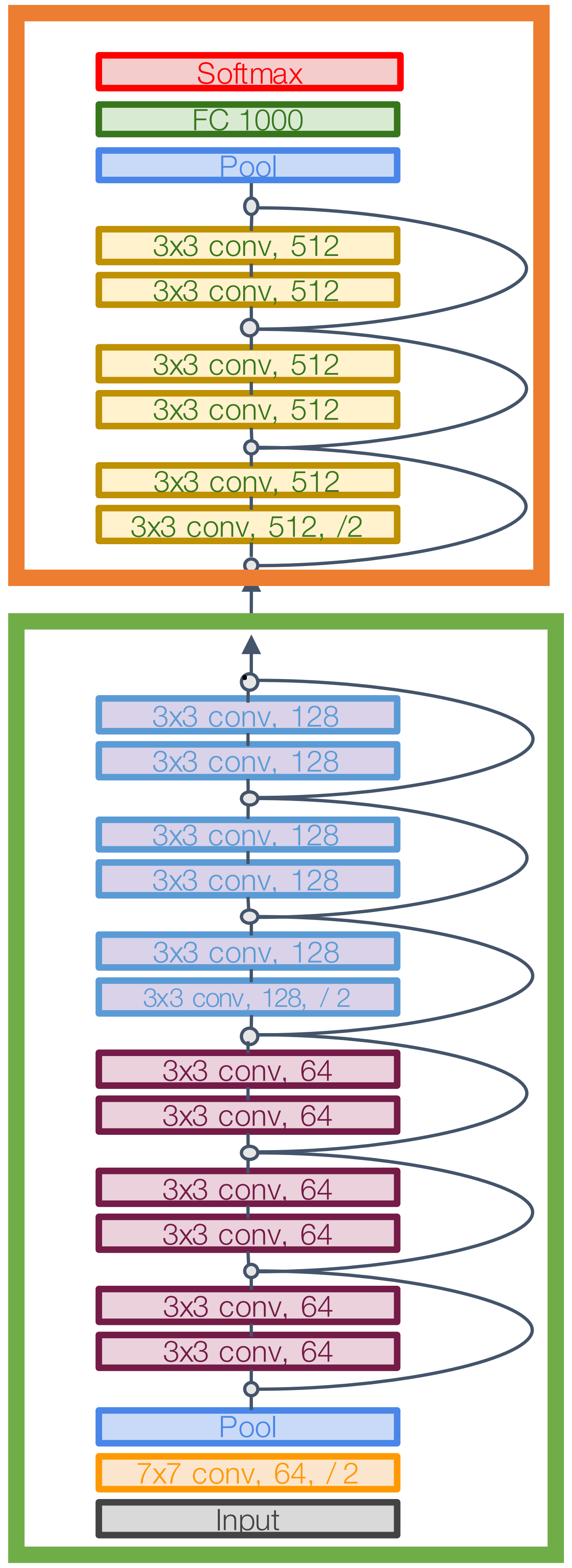
Image features

Run whole image through ConvNet

Input image

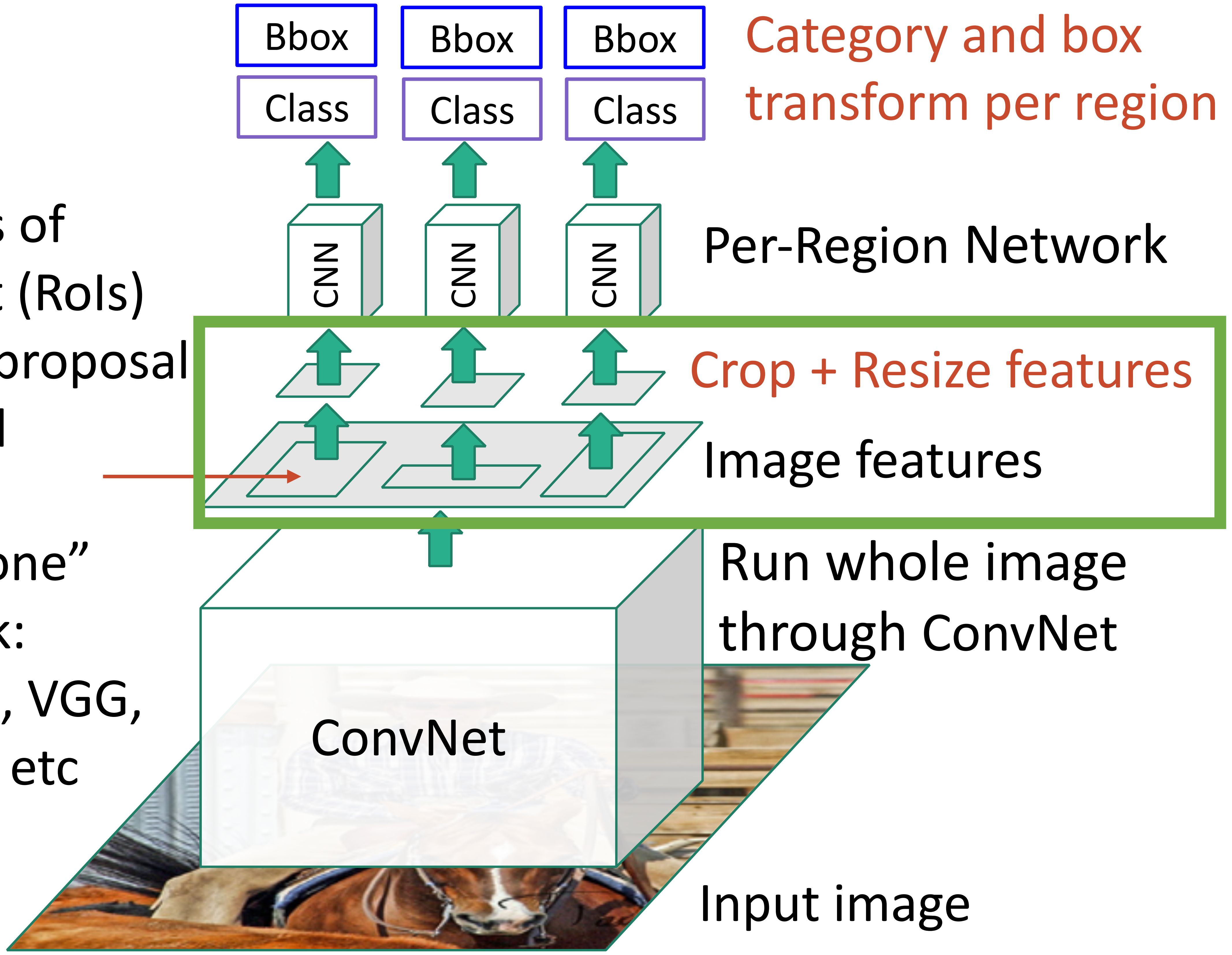
Regions of Interest (RoIs) from a proposal method

“Backbone” network: AlexNet, VGG, ResNet, etc



Example:
For ResNet, last stage is used as per-region network; the rest of the network is used as backbone

Fast R-CNN



Category and box transform per region

Per-Region Network

Regions of Interest (RoIs) from a proposal method

Crop + Resize features

Image features

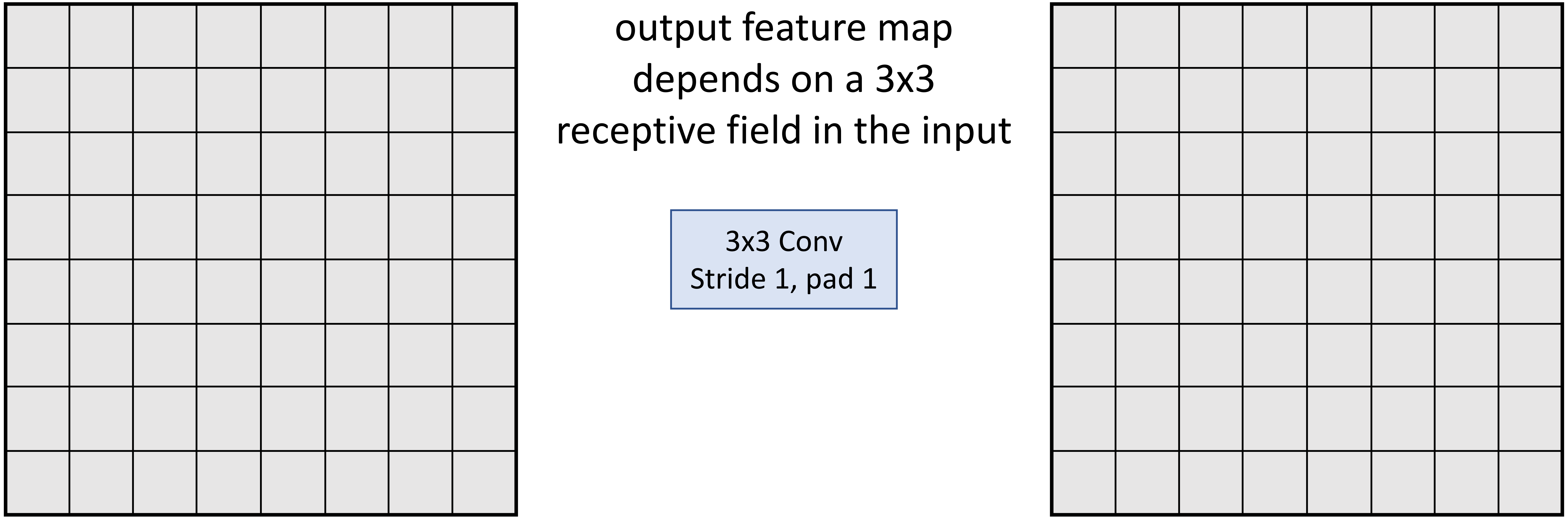
How to crop features?

"Backbone" network: AlexNet, VGG, ResNet, etc

Run whole image through ConvNet

Input image

Recall: Receptive Fields



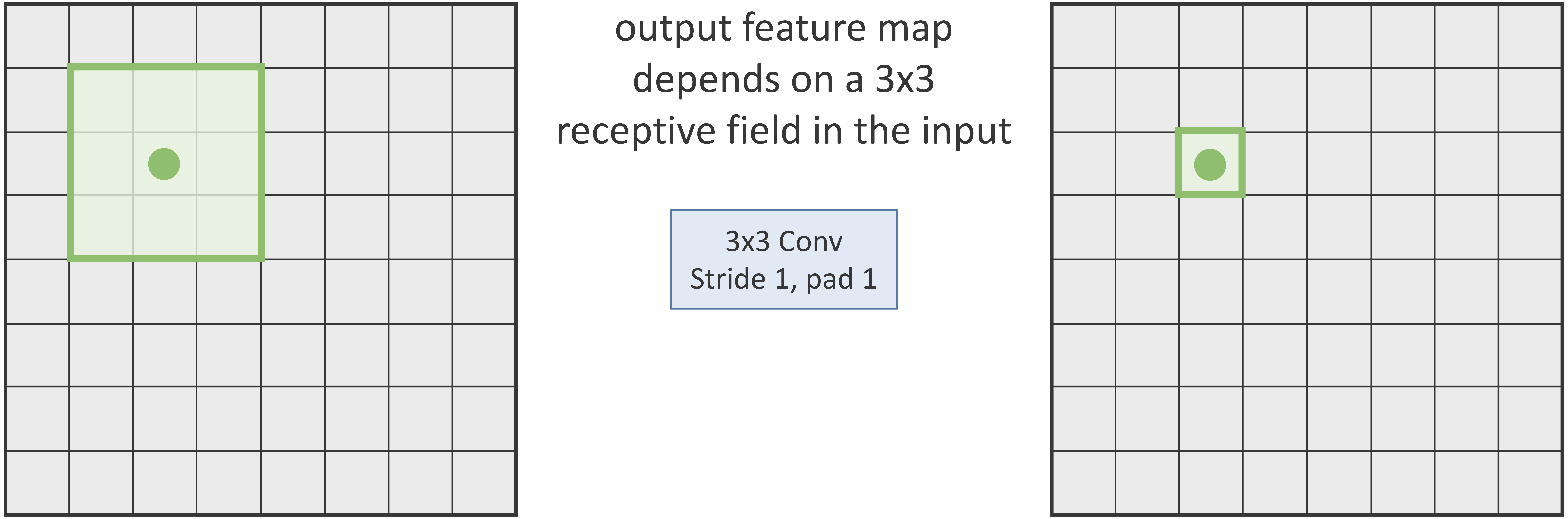
Every position in the output feature map depends on a 3x3 receptive field in the input

3x3 Conv
Stride 1, pad 1

Input Image: 8 x 8

Output Image: 8 x 8

Recall: Receptive Fields



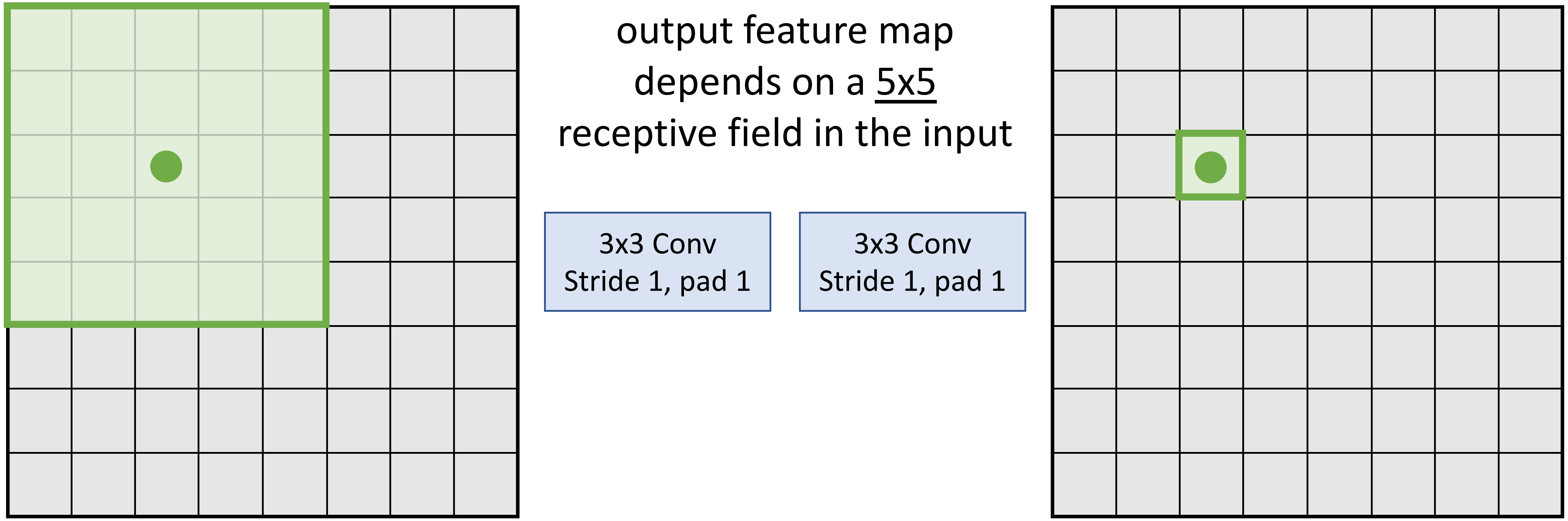
Every position in the output feature map depends on a 3x3 receptive field in the input

3x3 Conv
Stride 1, pad 1

Input Image: 8 x 8

Output Image: 8 x 8

Recall: Receptive Fields



Every position in the output feature map depends on a <u>5x5</u> receptive field in the input

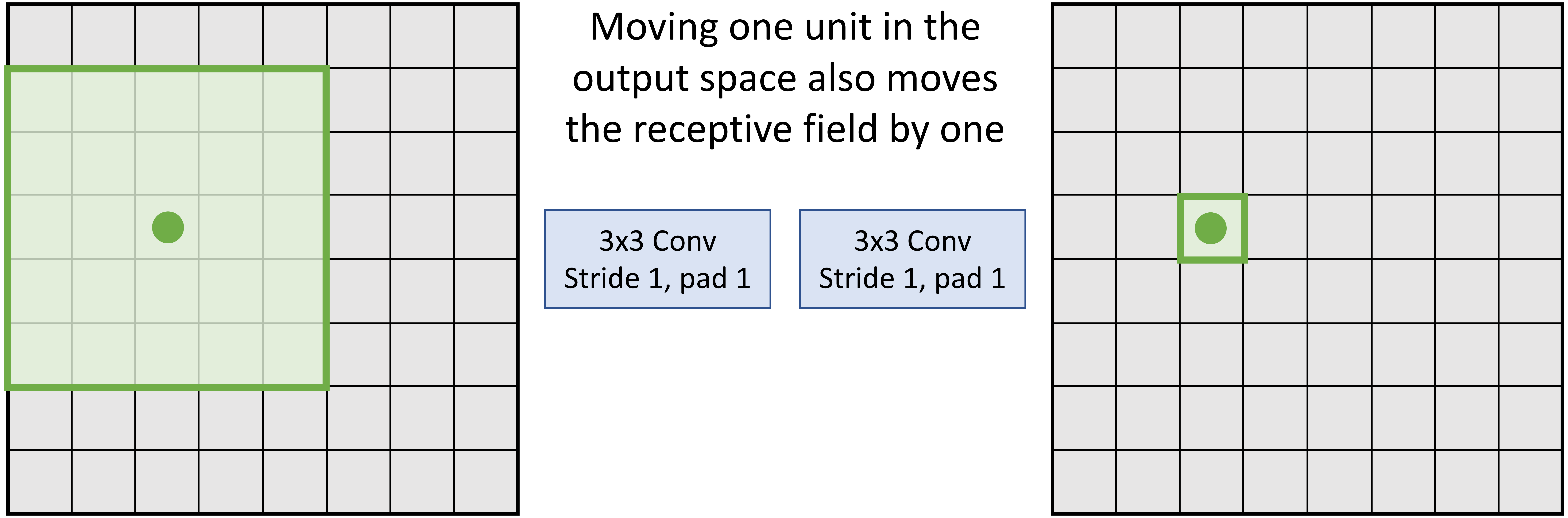
3x3 Conv
Stride 1, pad 1

3x3 Conv
Stride 1, pad 1

Input Image: 8 x 8

Output Image: 8 x 8

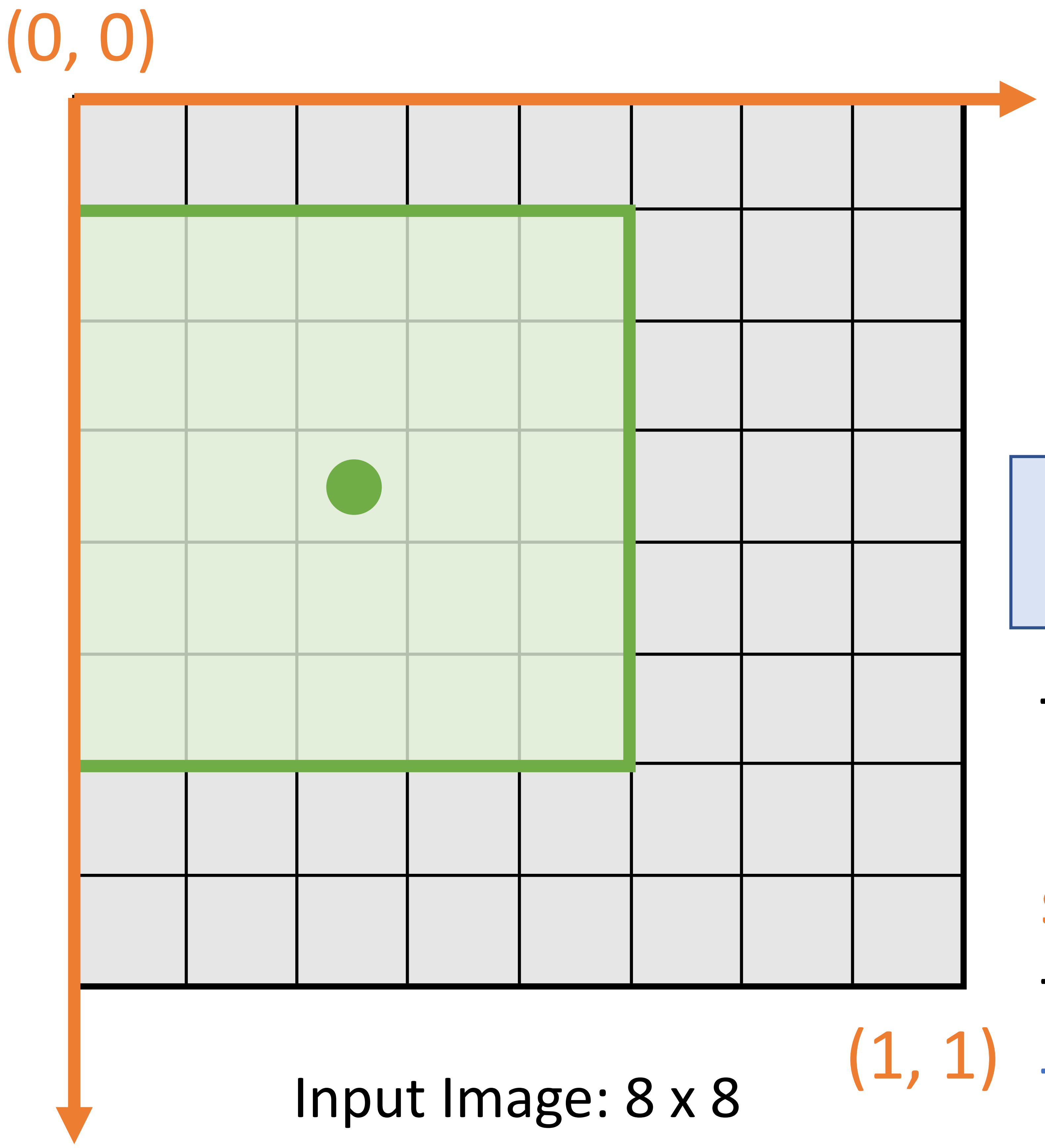
Recall: Receptive Fields


Moving one unit in the output space also moves the receptive field by one

Input Image: 8 x 8

Output Image: 8 x 8

Recall: Receptive Fields



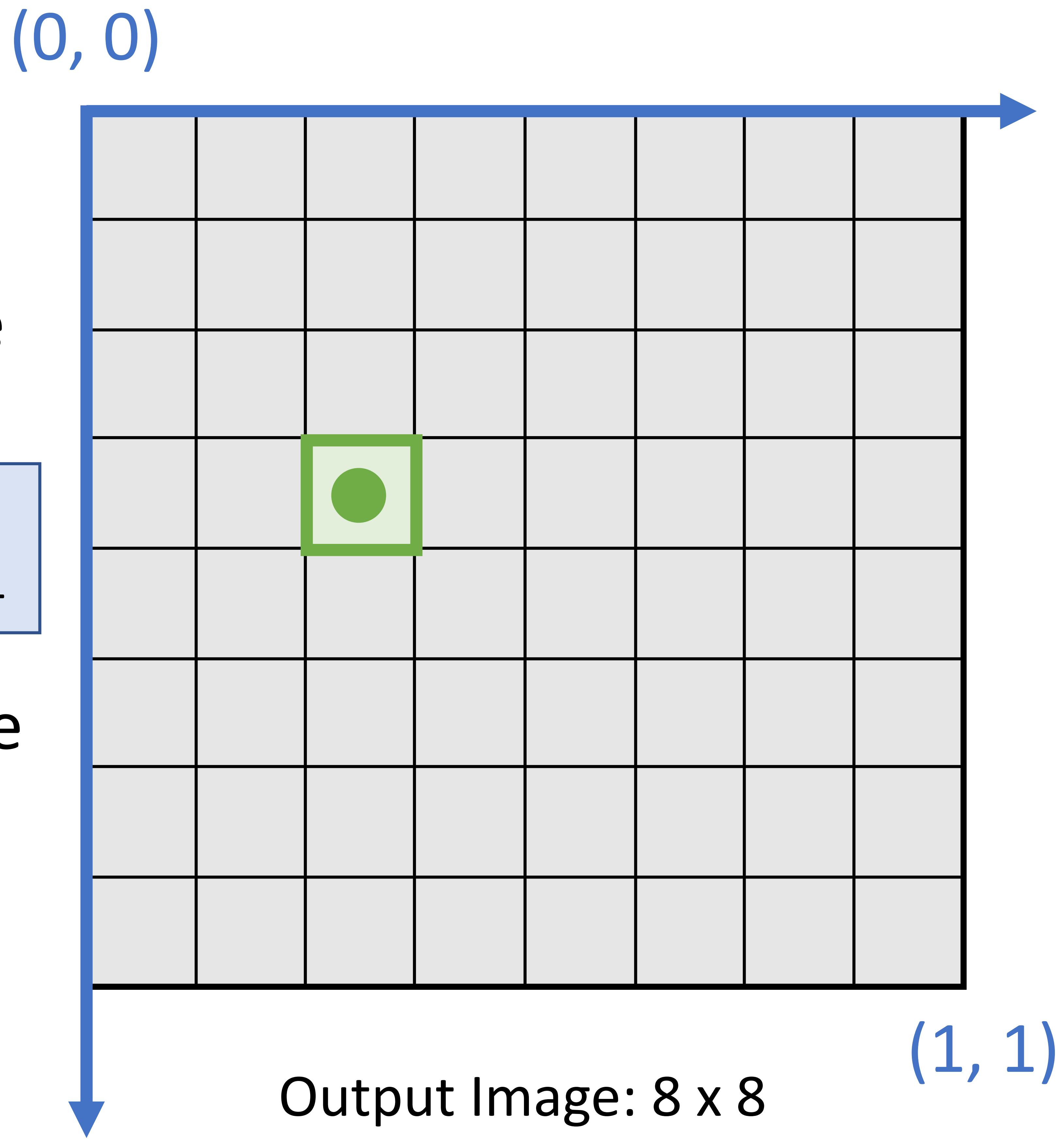
Input Image: 8 x 8

Moving one unit in the output space also moves the receptive field by one

3x3 Conv
Stride 1, pad 1

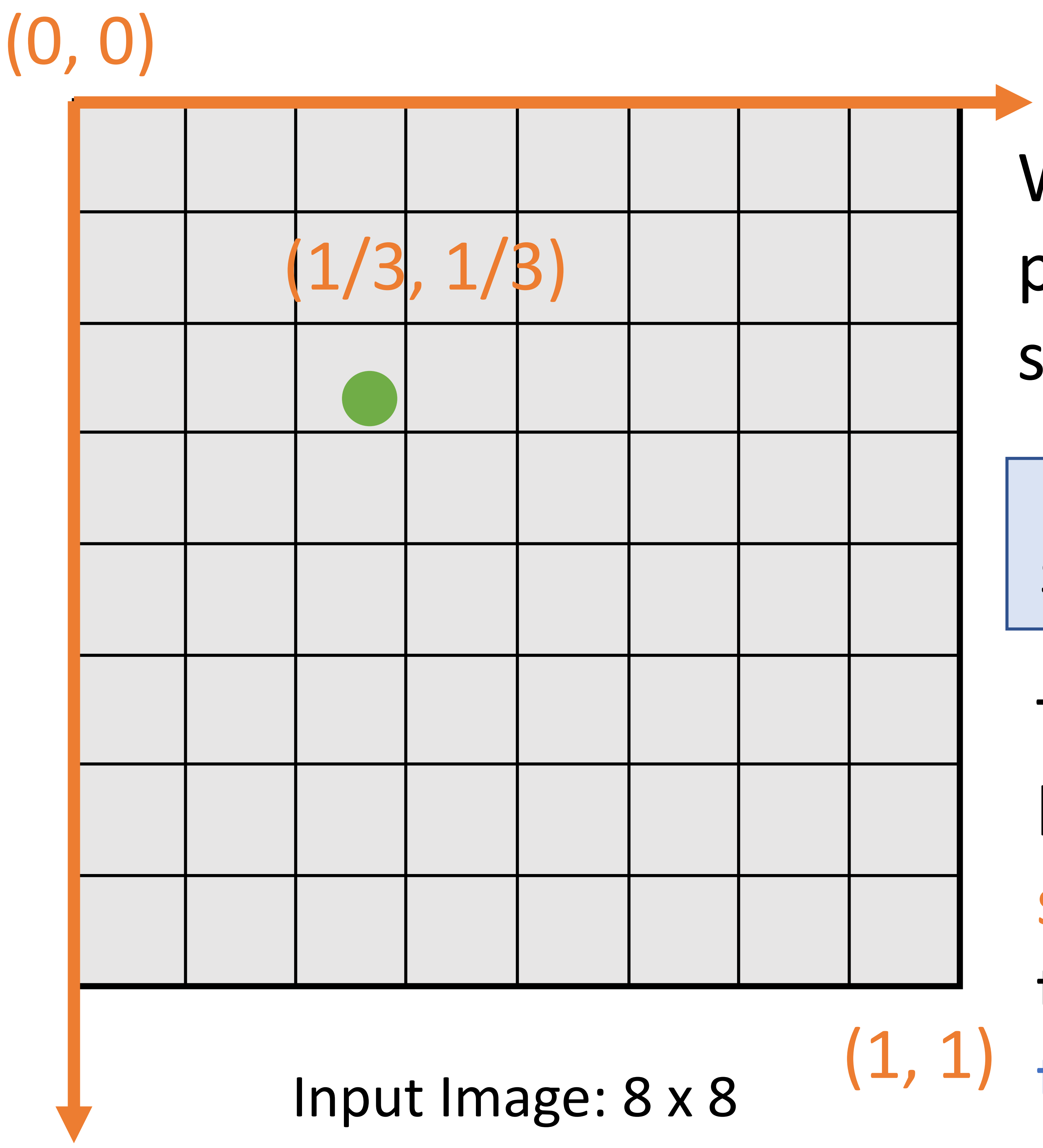
3x3 Conv
Stride 1, pad 1

There is a correspondence between the coordinate system of the input and the coordinate system of the output



Output Image: 8 x 8

Projecting Points



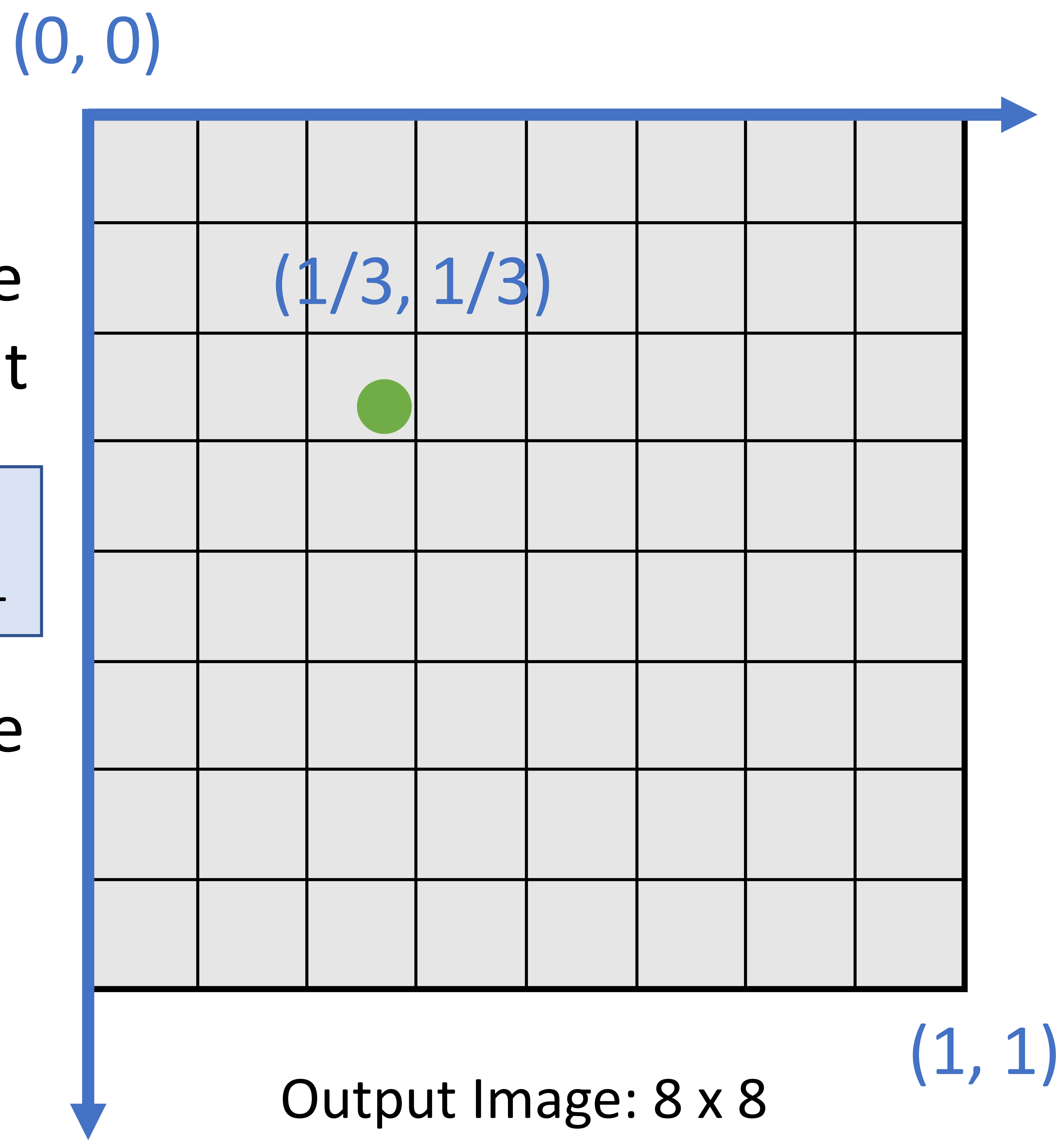
Input Image: 8 x 8

We can align arbitrary points between coordinate system of input and output

3x3 Conv
Stride 1, pad 1

3x3 Conv
Stride 1, pad 1

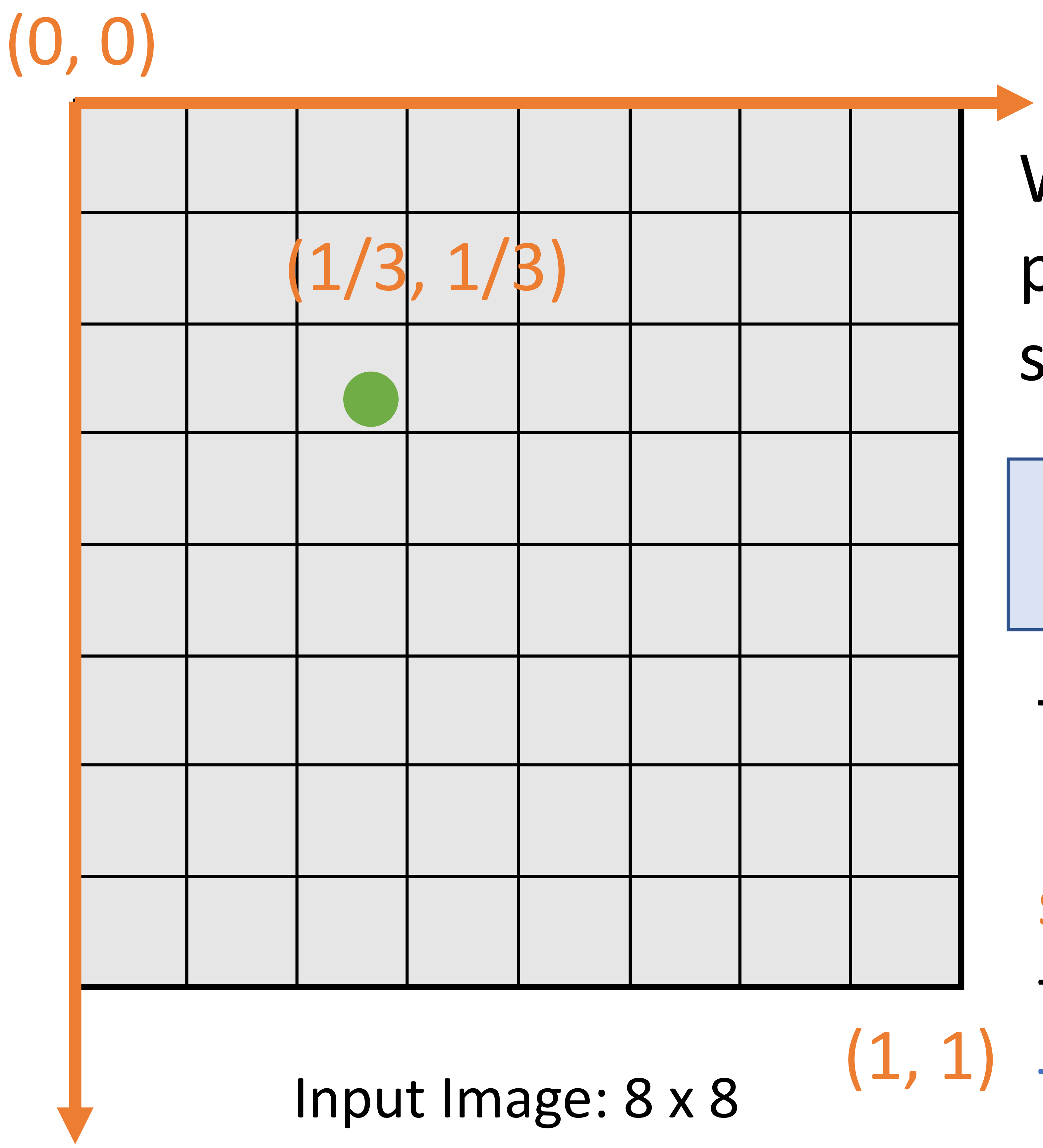
There is a correspondence between the coordinate system of the input and the coordinate system of the output



Output Image: 8 x 8

Projecting Points

Same logic holds for more complicated CNNs, even if spatial resolution of input and output are different



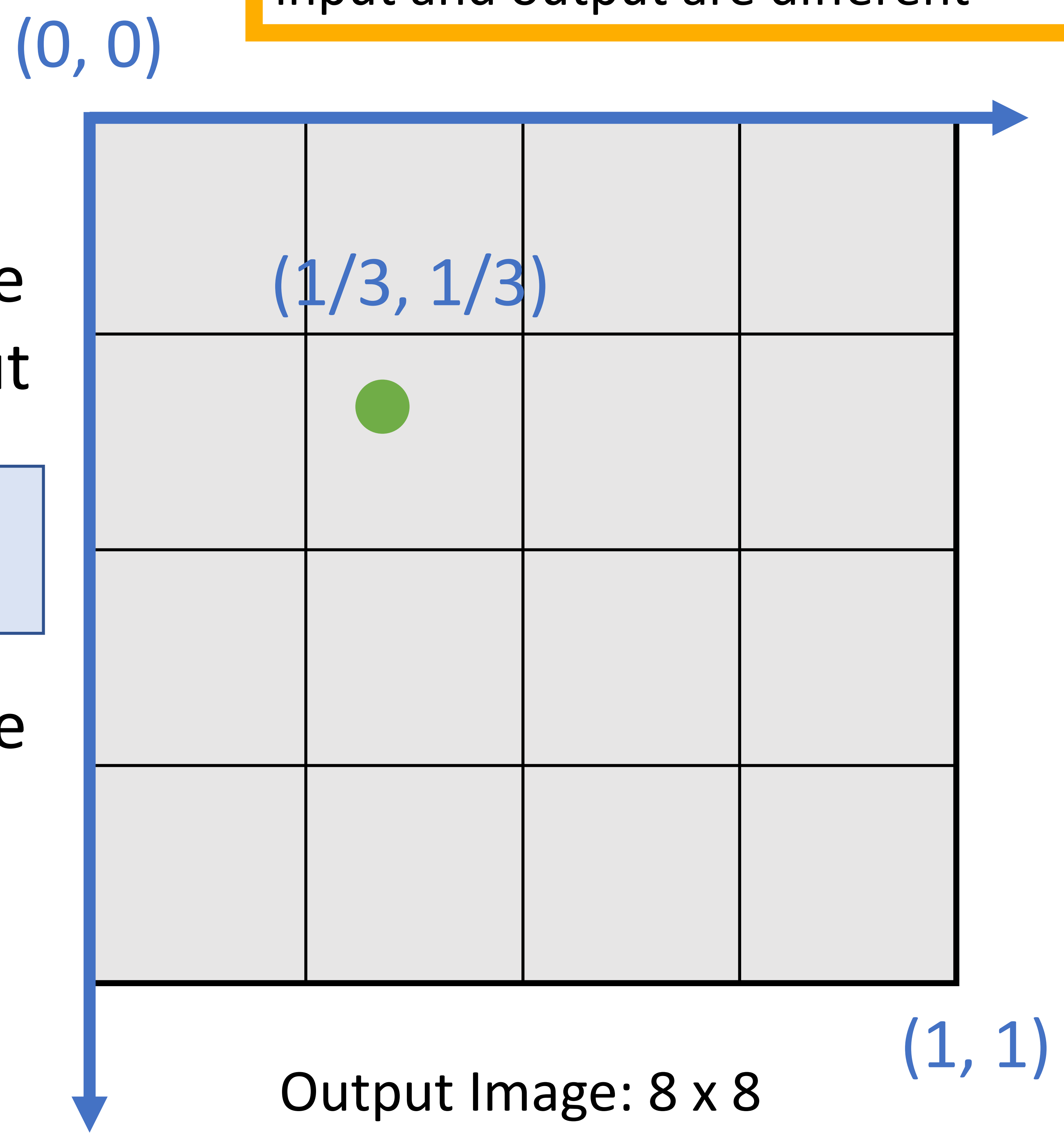
Input Image: 8 x 8

We can align arbitrary points between coordinate system of input and output

3x3 Conv
Stride 1, pad 1

2x2 MaxPool
Stride 2

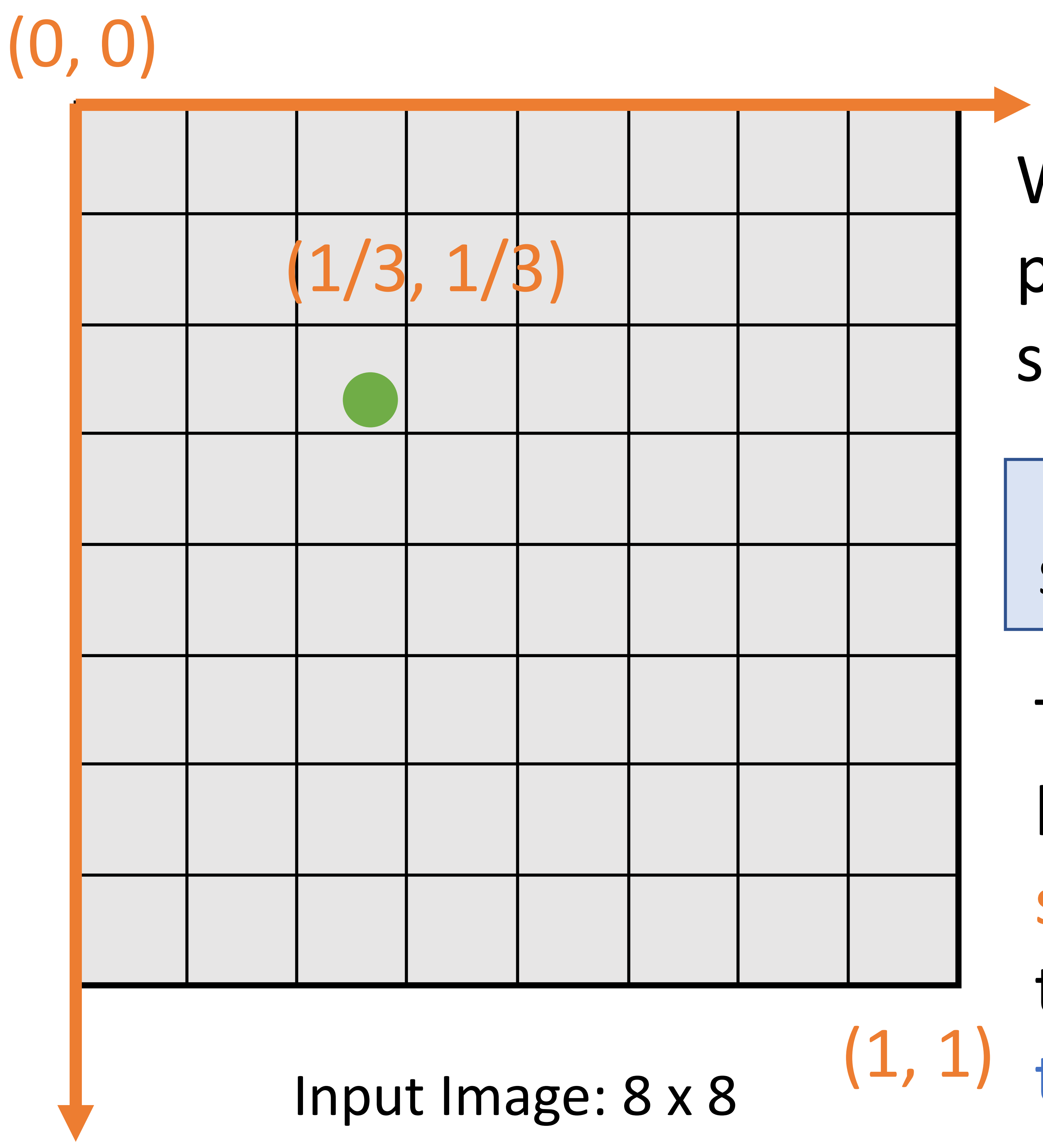
There is a correspondence between the coordinate system of the input and the coordinate system of the output



Output Image: 8 x 8

Projecting Points

Same logic holds for more complicated CNNs, even if spatial resolution of input and output are different



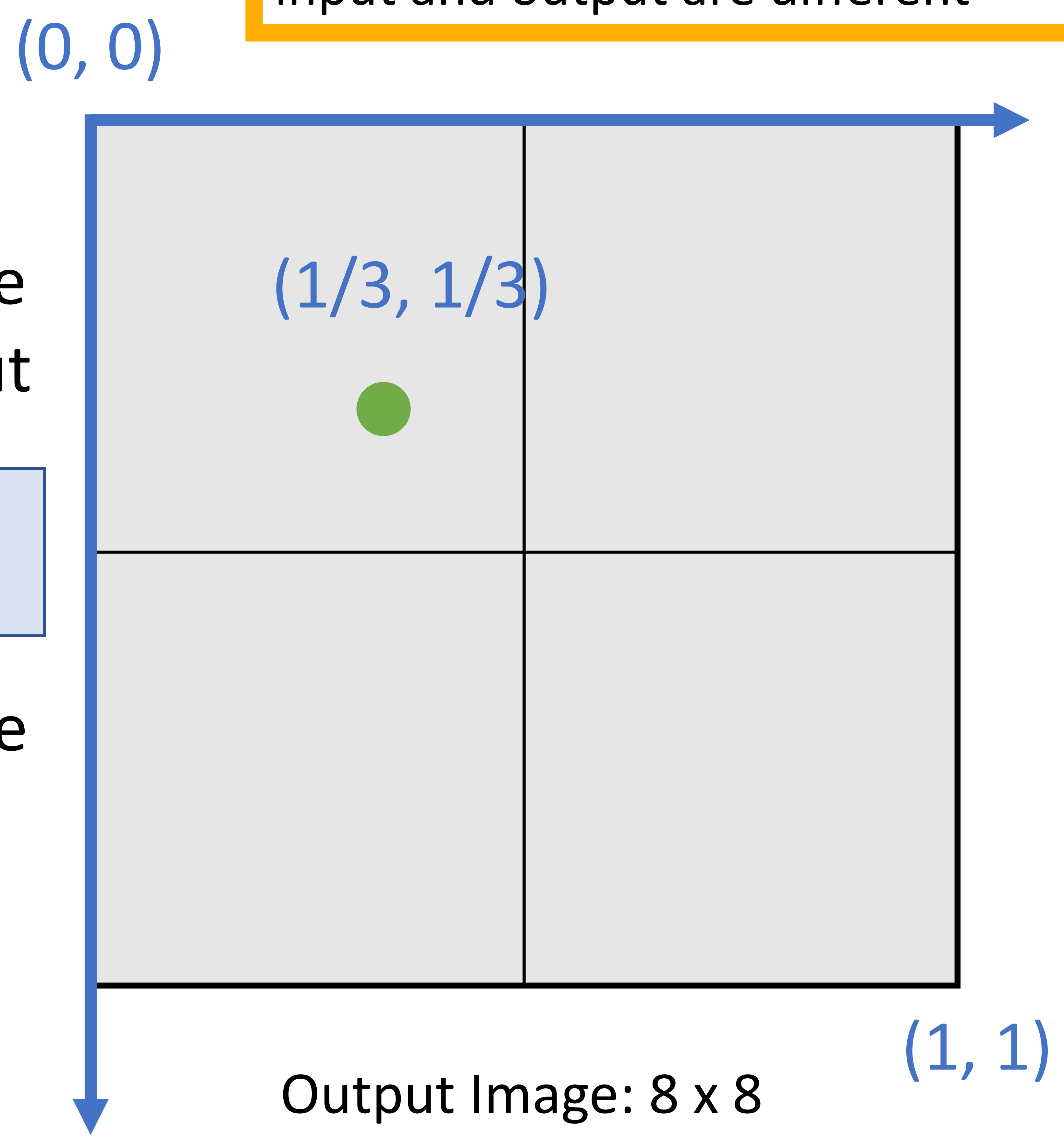
Input Image: 8 x 8

We can align arbitrary points between coordinate system of input and output

3x3 Conv
Stride 1, pad 1

4x4 MaxPool
Stride 4

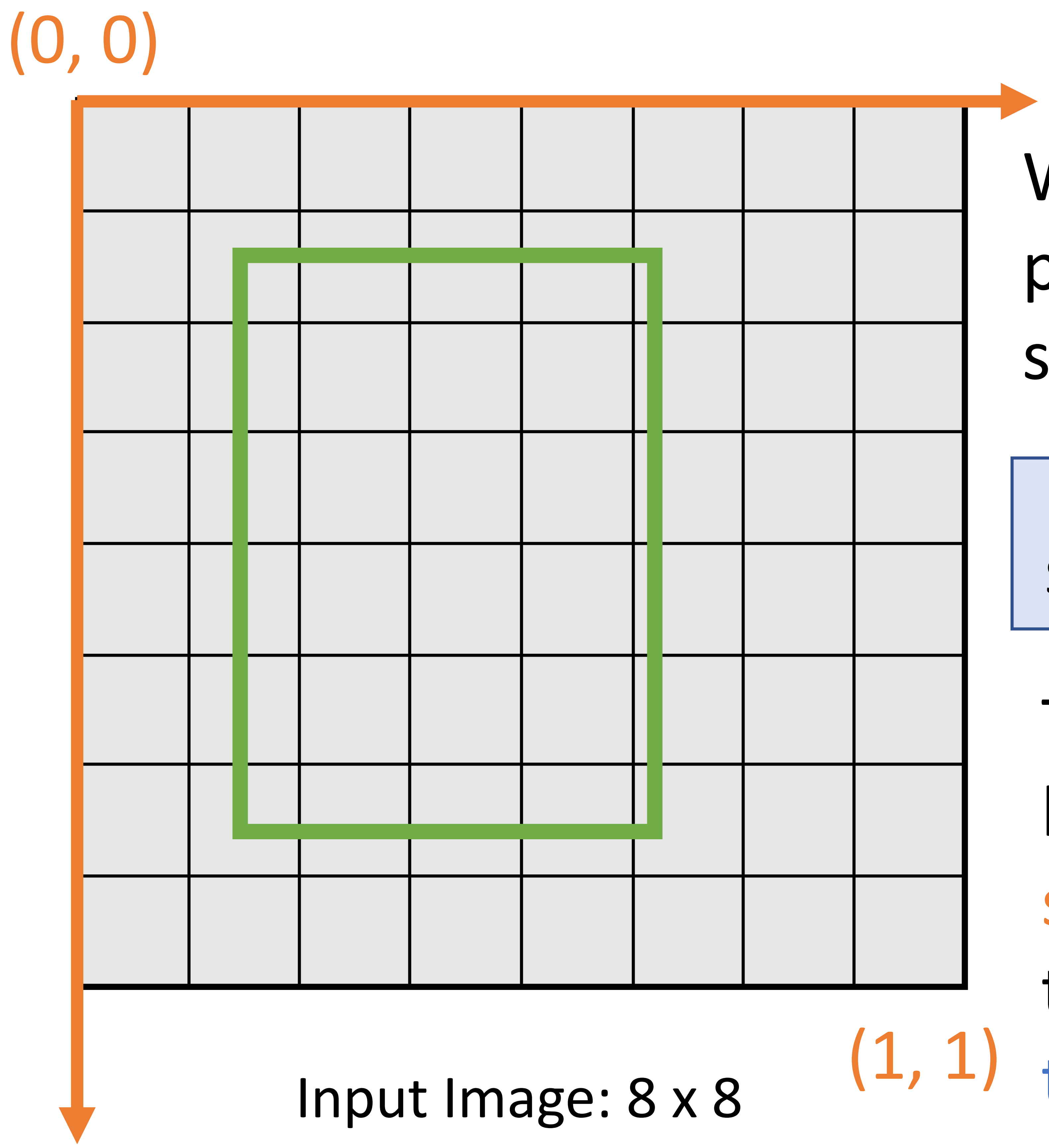
There is a correspondence between the coordinate system of the input and the coordinate system of the output



Output Image: 8 x 8

Projecting Points

We can use this idea to project **bounding boxes** between an input image and a feature map

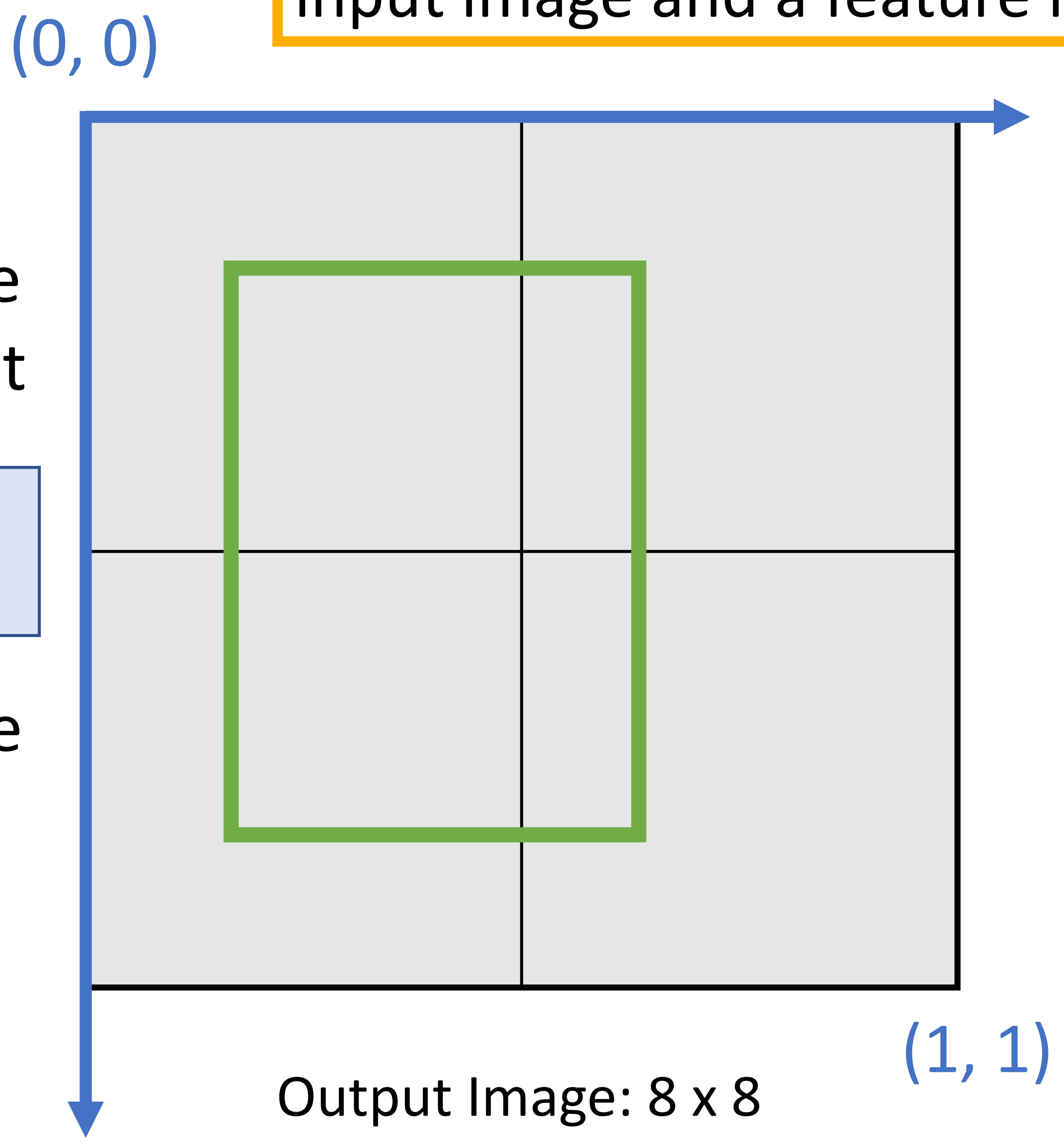


We can align arbitrary points between coordinate system of input and output

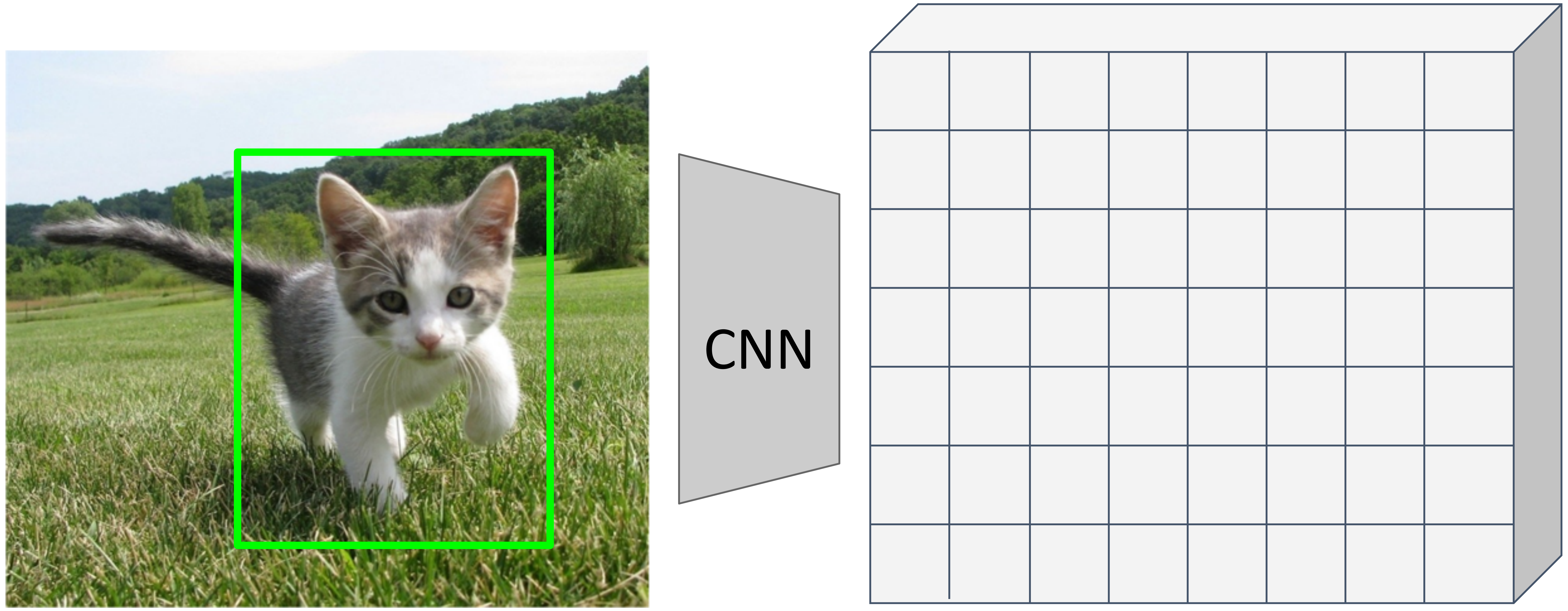
3x3 Conv
Stride 1, pad 1

4x4 MaxPool
Stride 4

There is a correspondence between the coordinate system of the input and the coordinate system of the output



Cropping Features: RoI Pool



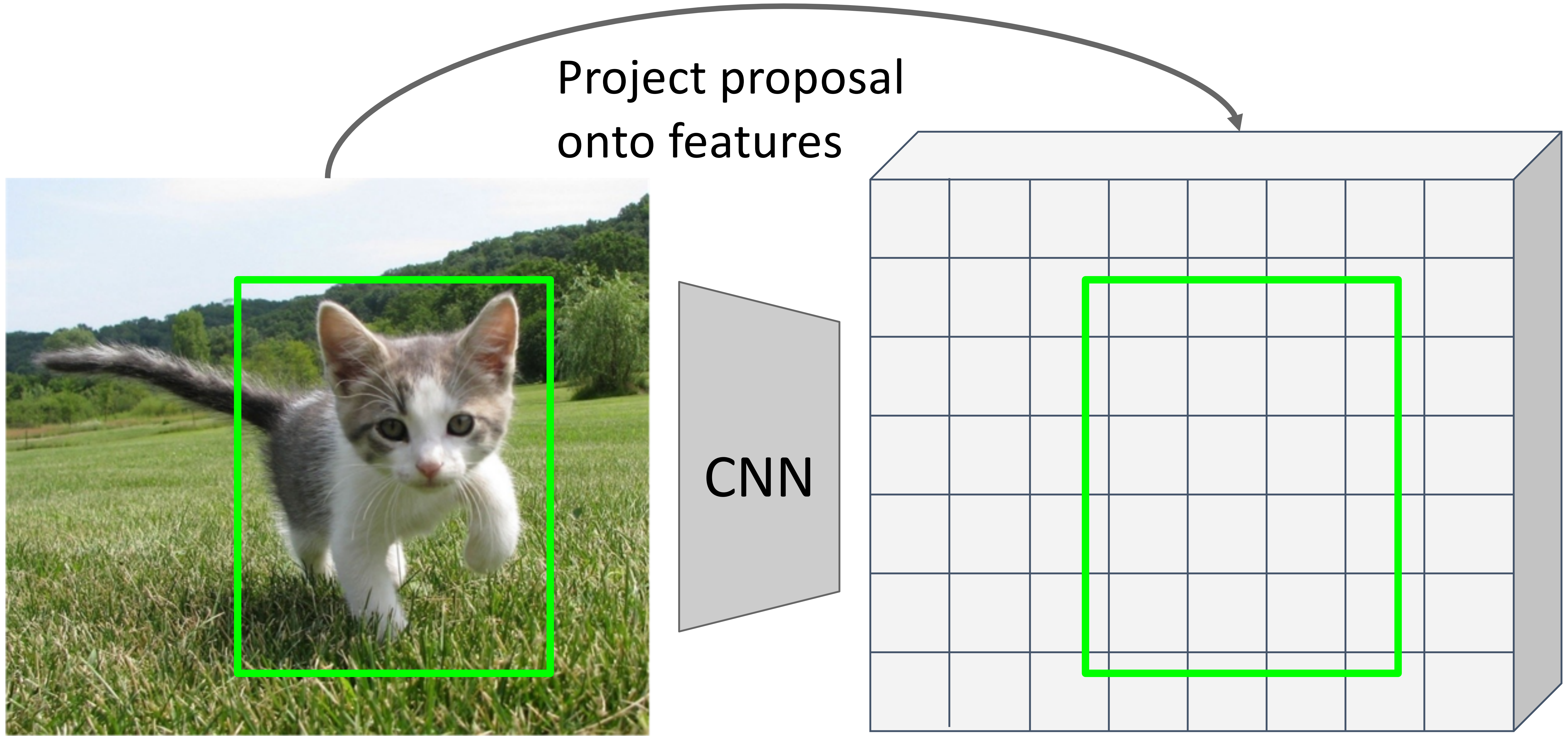
Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

Want features for the
box of a fixed size
(2x2 in this example,
7x7 or 14x14 in practice)

Girshick, “Fast R-CNN”, ICCV 2015.

Cropping Features: RoI Pool



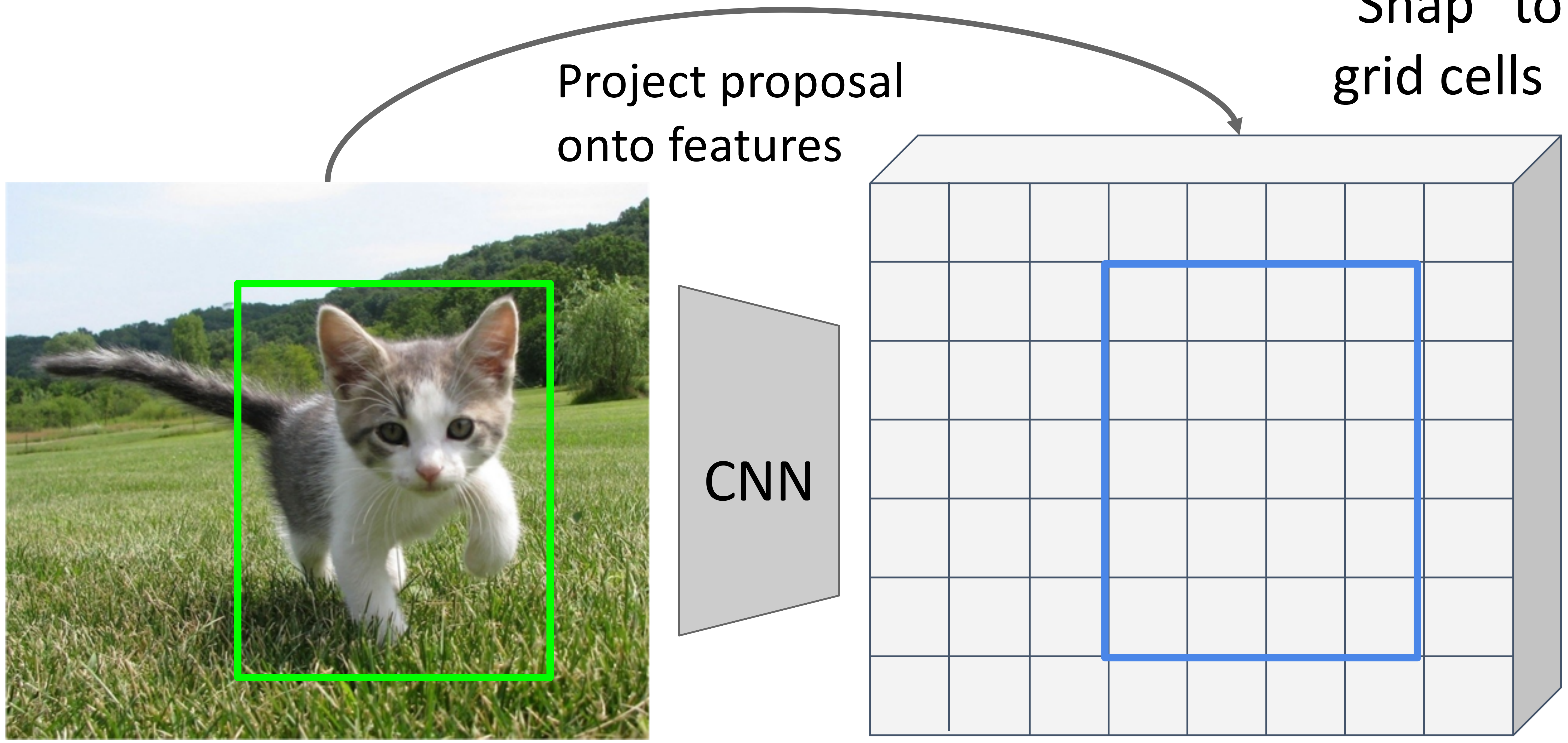
Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

Want features for the
box of a fixed size
(2x2 in this example,
7x7 or 14x14 in practice)

Girshick, “Fast R-CNN”, ICCV 2015.

Cropping Features: RoI Pool



“Snap” to grid cells

Project proposal onto features

CNN

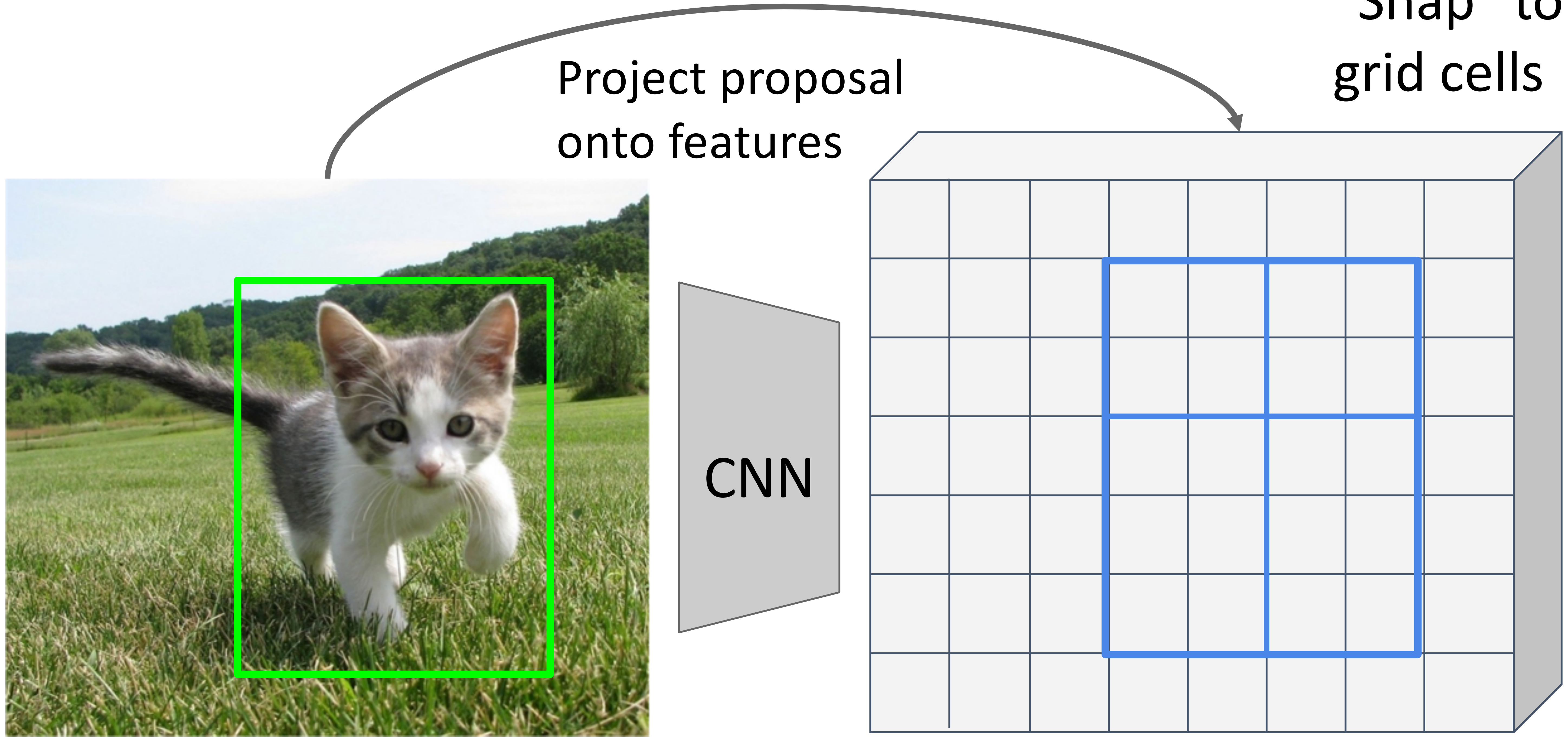
Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

Want features for the box of a fixed size (2x2 in this example, 7x7 or 14x14 in practice)

Girshick, “Fast R-CNN”, ICCV 2015.

Cropping Features: RoI Pool



"Snap" to grid cells

Divide into 2x2 grid of (roughly) equal subregions

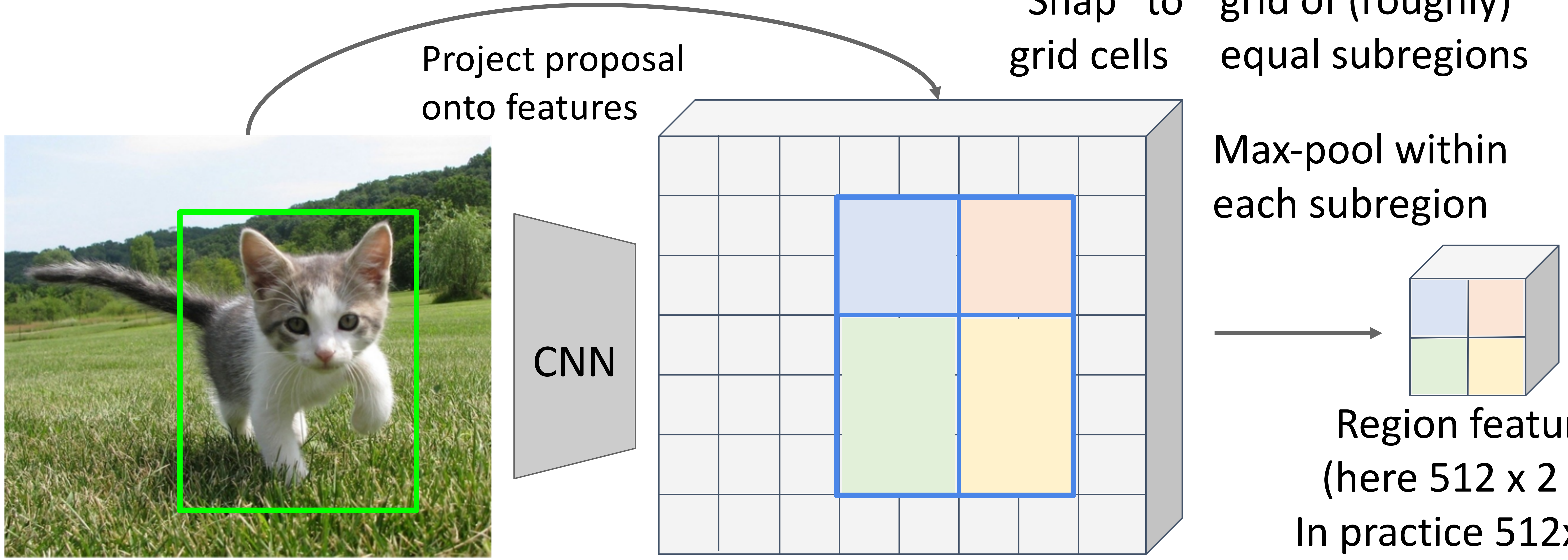
Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

Want features for the box of a fixed size (2x2 in this example, 7x7 or 14x14 in practice)

Girshick, "Fast R-CNN", ICCV 2015.

Cropping Features: RoI Pool



Project proposal onto features

CNN

Input Image

(e.g. 3 x 640 x 480)

Image features

(e.g. 512 x 20 x 15)

“Snap” to grid cells

Divide into 2x2 grid of (roughly) equal subregions

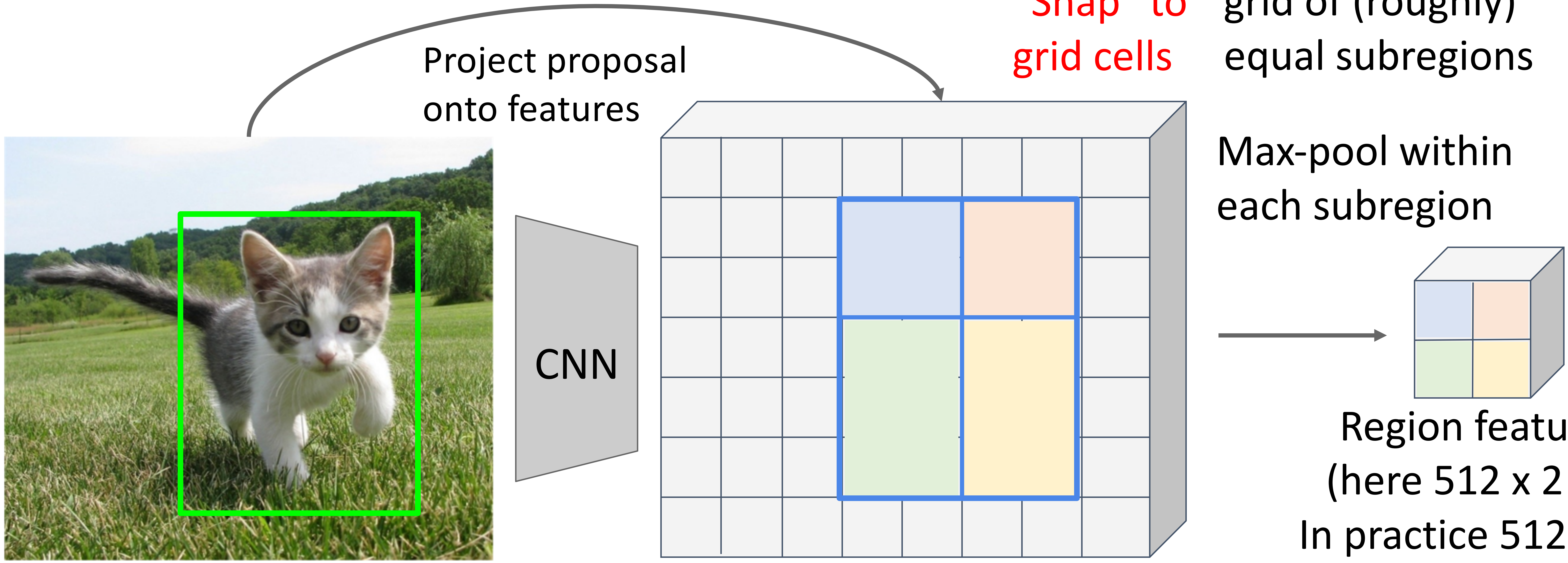
Max-pool within each subregion

Region features (here 512 x 2 x 2; In practice 512x7x7)

Region features always the same size even if input regions have different sizes!

Girshick, “Fast R-CNN”, ICCV 2015.

Cropping Features: RoI Pool



Project proposal onto features

CNN

"Snap" to grid cells

Divide into 2x2 grid of (roughly) equal subregions

Max-pool within each subregion

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

Region features
(here 512 x 2 x 2;
In practice 512x7x7)

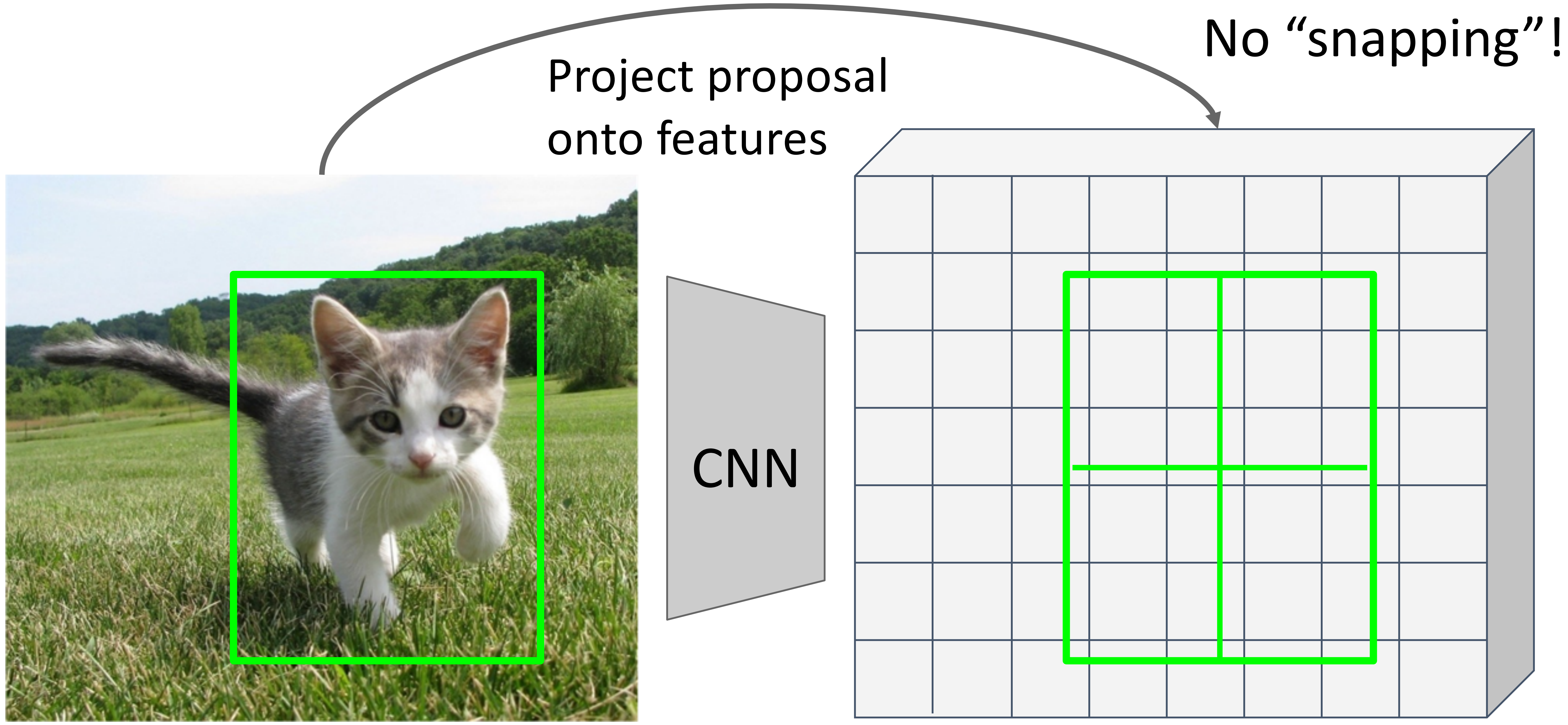
Region features always the same size even if input regions have different sizes!

Problem: Slight misalignment due to snapping; different-sized subregions is weird

Girshick, "Fast R-CNN", ICCV 2015.

Cropping Features: RoI Align

Divide into equal-sized subregions
(may not be aligned to grid!)



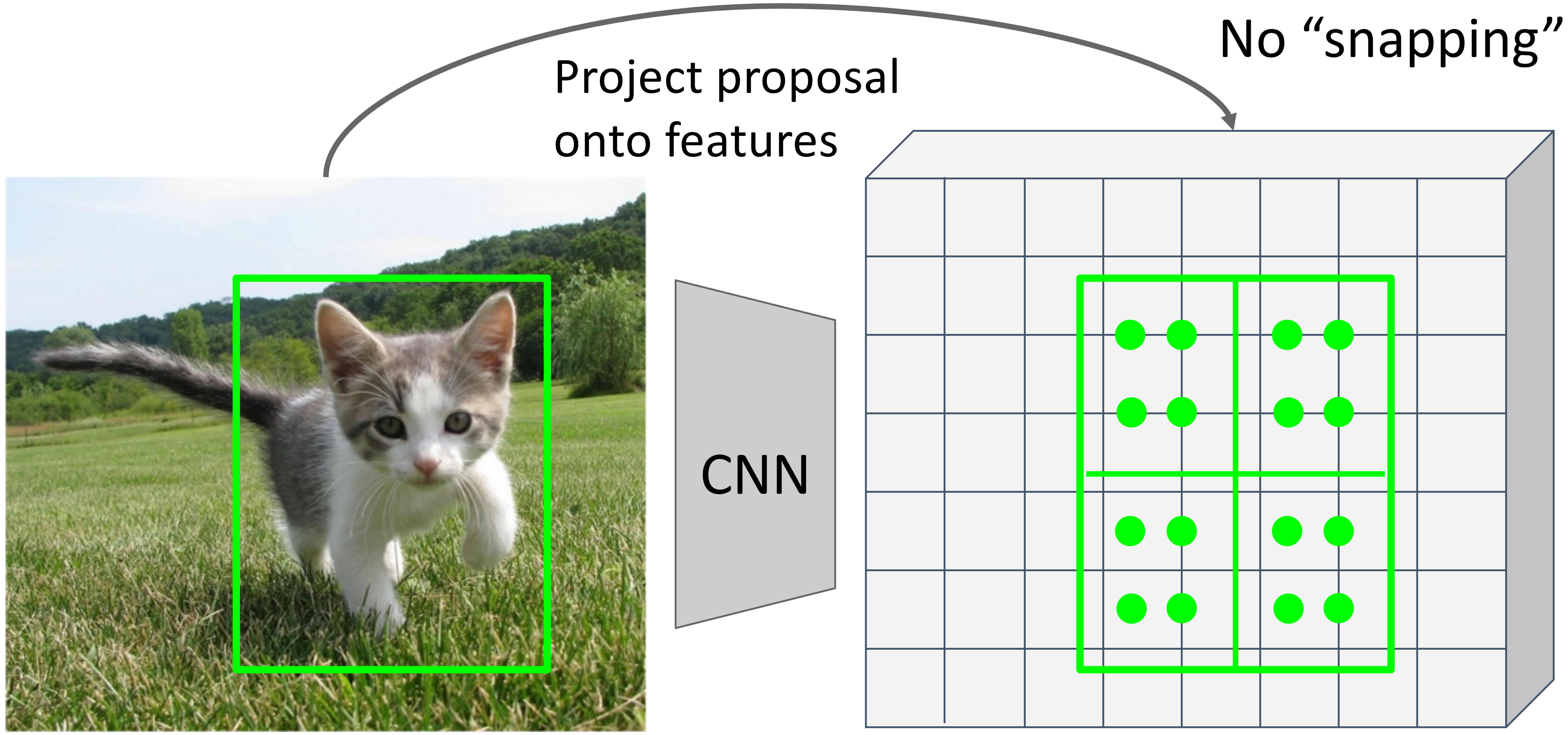
Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

Want features for the
box of a fixed size
(2x2 in this example,
7x7 or 14x14 in practice)

Girshick, "Fast R-CNN", ICCV 2015.

Cropping Features: RoI Align



Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

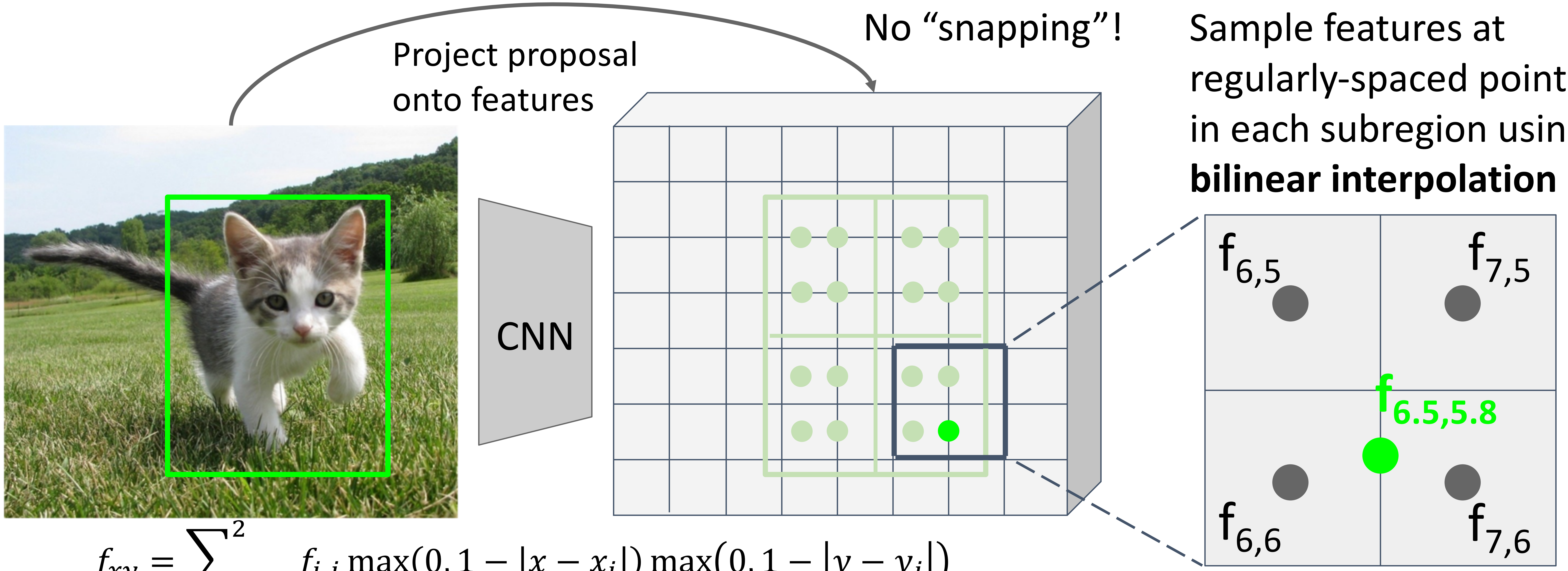
Divide into equal-sized subregions (may not be aligned to grid!)

Sample features at regularly-spaced points in each subregion using **bilinear interpolation**

He et al, "Mask R-CNN", ICCV 2017

Cropping Features: RoI Align

Divide into equal-sized subregions (may not be aligned to grid!)



Sample features at regularly-spaced points in each subregion using **bilinear interpolation**

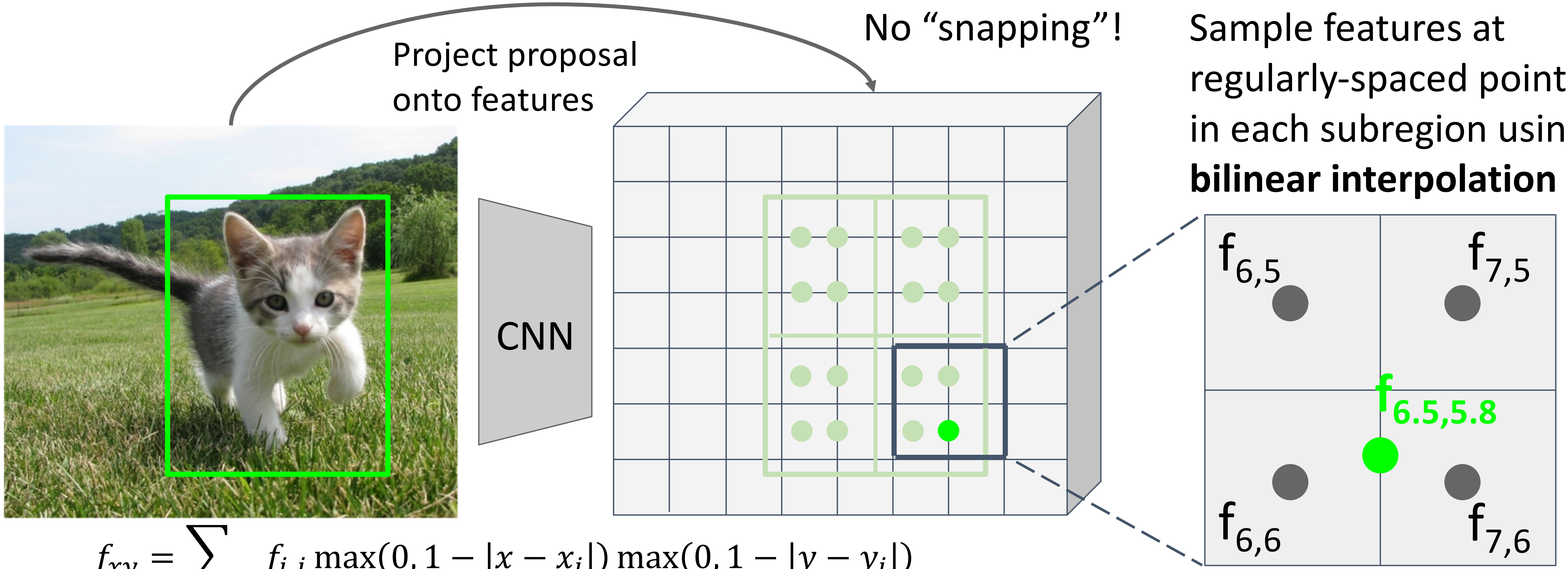
Feature f_{xy} for point (x, y) is a linear combination of features at its four neighboring grid cells:

$$f_{xy} = \sum_{i,j=1}^{2} f_{i,j} \max(0, 1 - |x - x_i|) \max(0, 1 - |y - y_j|)$$

He et al, “Mask R-CNN”, ICCV 2017

Cropping Features: RoI Align

Divide into equal-sized subregions (may not be aligned to grid!)



Project proposal onto features

No “snapping”!

Sample features at regularly-spaced points in each subregion using **bilinear interpolation**

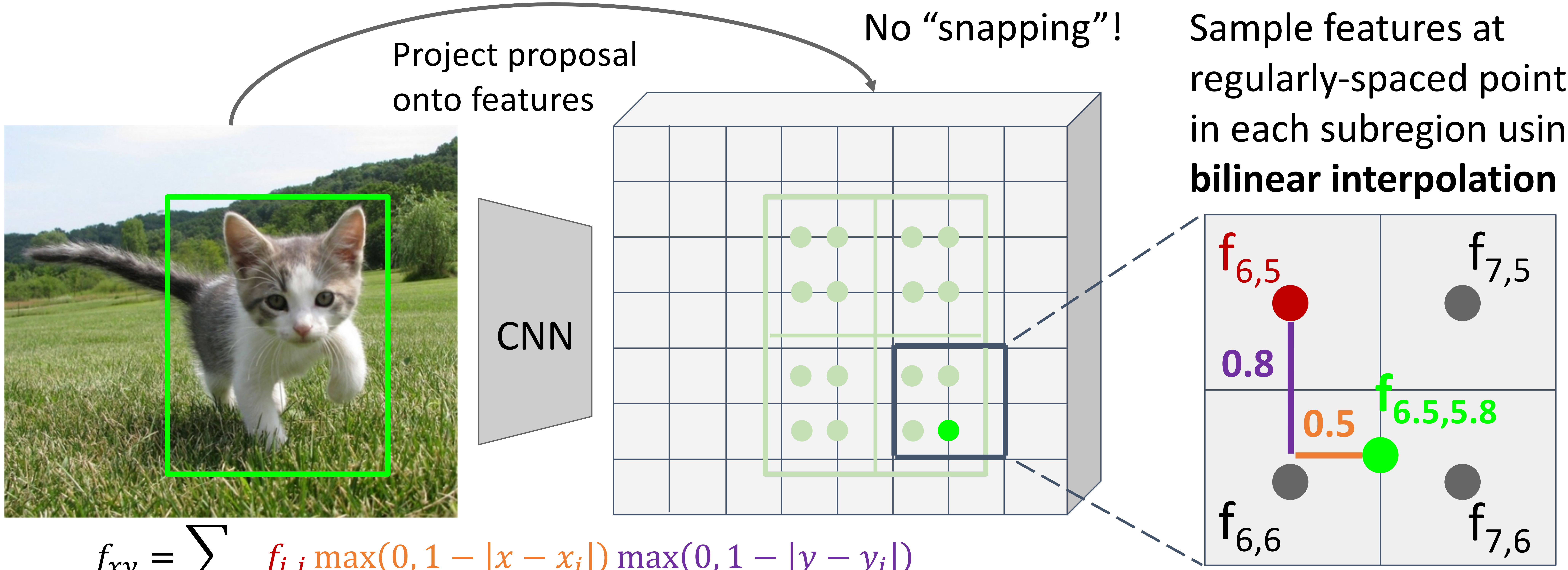
$$f_{xy} = \sum_{i,j} f_{i,j} \max(0, 1-|x-x_i|)\max(0, 1-|y-y_i|)$$

$$f_{6.5,5.8} = (f_{6,5} * 0.5 * 0.2) + (f_{7,5} * 0.5 * 0.2) + (f_{6,6} * 0.5 * 0.8) + (f_{7,6} * 0.5 * 0.8)$$

Feature f_{xy} for point (x, y) is a linear combination of features at its four neighboring grid cells:

He et al, “Mask R-CNN”, ICCV 2017

Cropping Features: RoI Align



Project proposal onto features

CNN

No “snapping”!

Sample features at regularly-spaced points in each subregion using **bilinear interpolation**

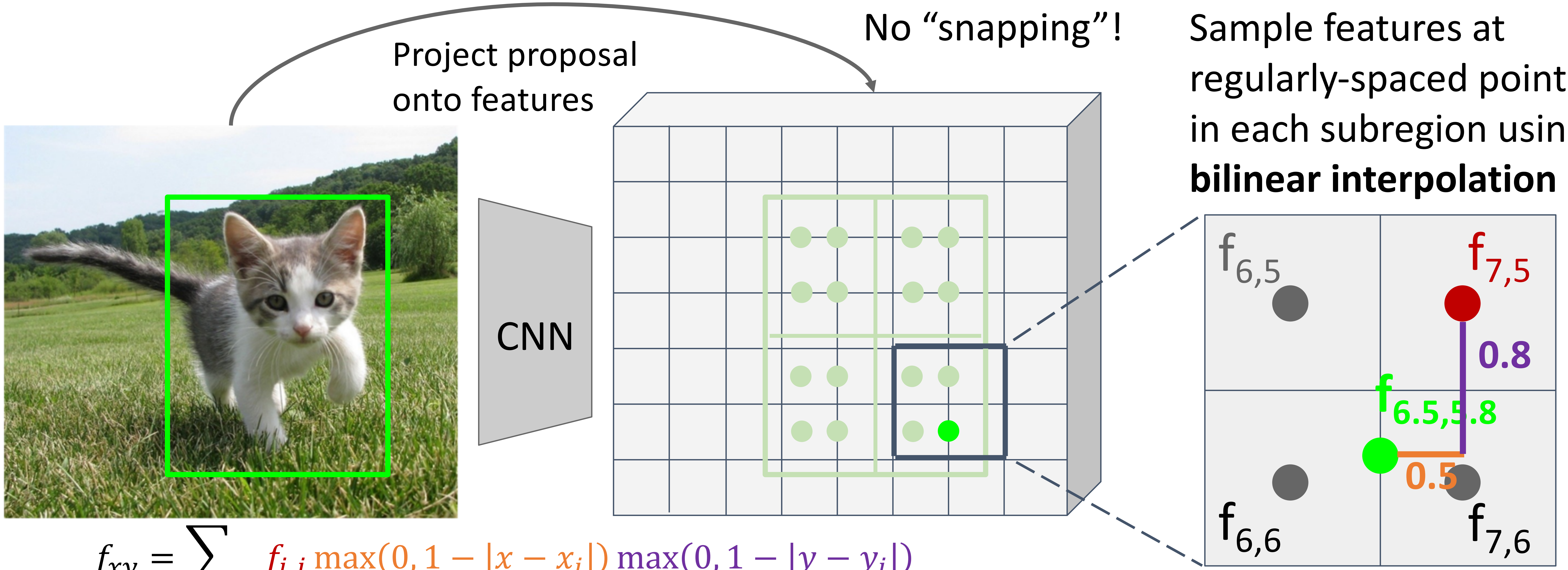
$$f_{xy} = \sum_{i,j} f_{i,j} \max(0, 1 - |x - x_i|) \max(0, 1 - |y - y_i|)$$

$$f_{6.5,5.8} = (f_{6,5} * 0.5 * 0.2) + (f_{7,5} * 0.5 * 0.2) + (f_{6,6} * 0.5 * 0.8) + (f_{7,6} * 0.5 * 0.8)$$

Feature f_{xy} for point (x, y) is a linear combination of features at its four neighboring grid cells:

He et al, “Mask R-CNN”, ICCV 2017

Cropping Features: RoI Align



Project proposal onto features

No “snapping”!

Sample features at regularly-spaced points in each subregion using **bilinear interpolation**

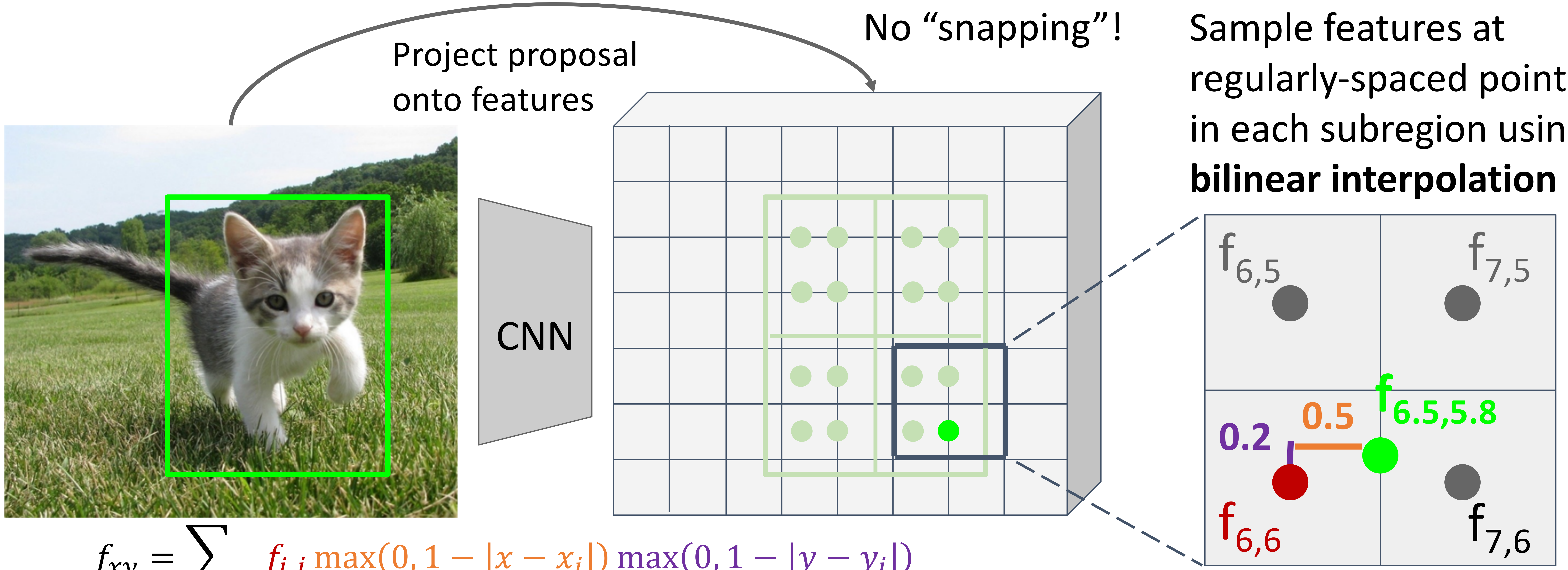
Feature f_{xy} for point (x, y) is a linear combination of features at its four neighboring grid cells:

$$f_{xy} = \sum_{i,j} f_{i,j} \max(0, 1 - |x - x_i|) \max(0, 1 - |y - y_i|)$$

$$f_{6.5,5.8} = (f_{6,5} * 0.5 * 0.2) + (f_{7,5} * 0.5 * 0.2) + (f_{6,6} * 0.5 * 0.8) + (f_{7,6} * 0.5 * 0.8)$$

He et al, “Mask R-CNN”, ICCV 2017

Cropping Features: RoI Align



Project proposal onto features

No “snapping”!

Sample features at regularly-spaced points in each subregion using **bilinear interpolation**

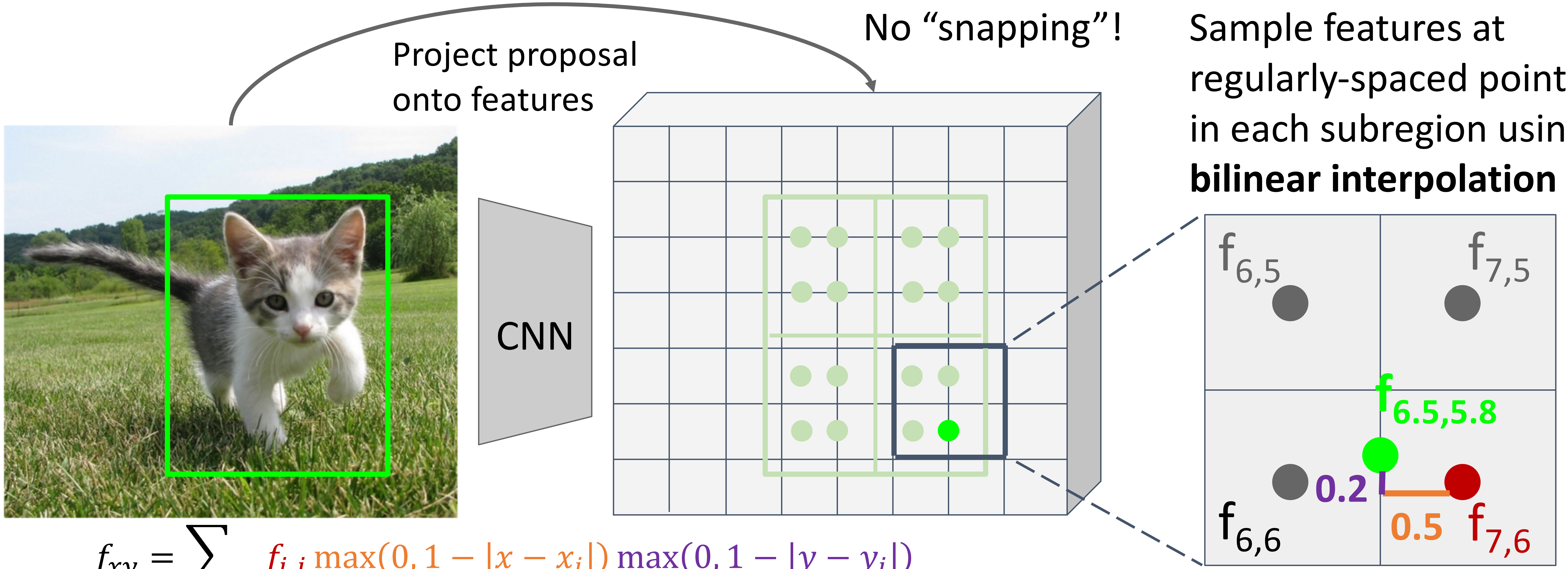
$$f_{xy} = \sum_{i,j} f_{i,j} \max(0, 1-|x-x_i|) \max(0, 1-|y-y_i|)$$

$$f_{6.5,5.8} = (f_{6,5} * 0.5 * 0.2) + (f_{7,5} * 0.5 * 0.2) + (f_{6,6} * 0.5 * 0.8) + (f_{7,6} * 0.5 * 0.8)$$

Feature f_{xy} for point (x, y) is a linear combination of features at its four neighboring grid cells:

He et al, “Mask R-CNN”, ICCV 2017

Cropping Features: RoI Align



Project proposal onto features

No “snapping”!

Sample features at regularly-spaced points in each subregion using **bilinear interpolation**

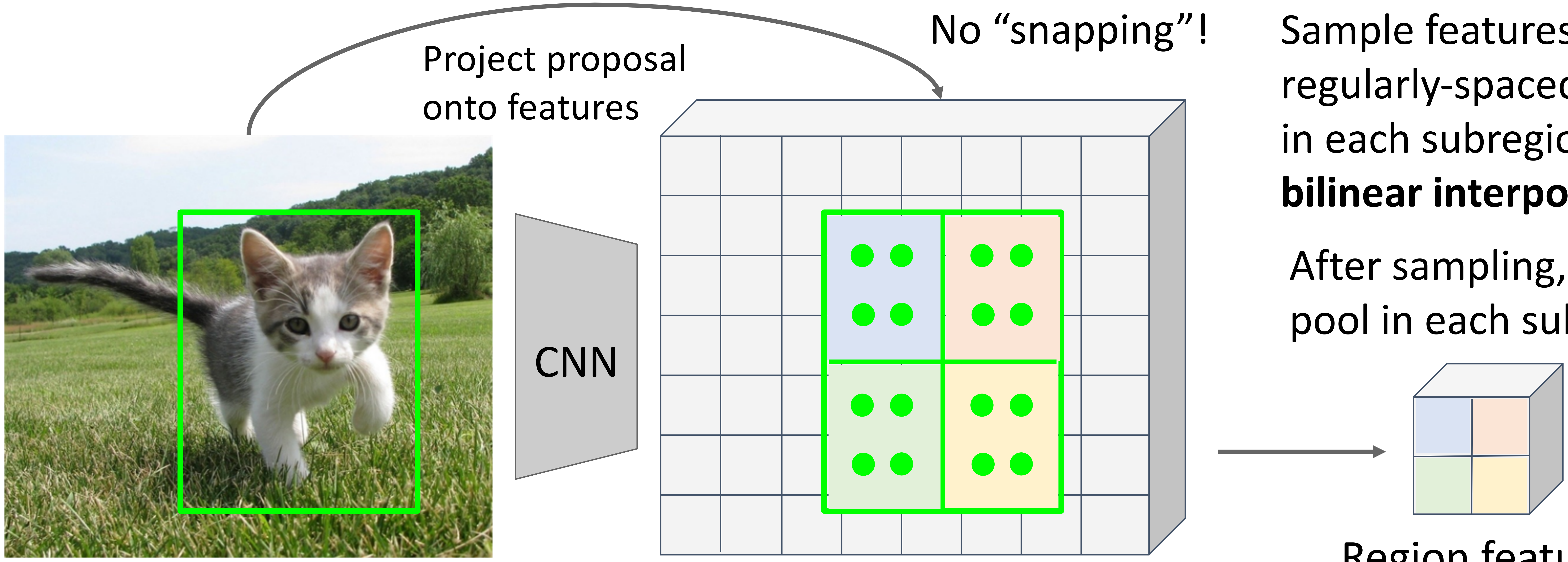
Feature f_{xy} for point (x, y) is a linear combination of features at its four neighboring grid cells:

$$f_{xy} = \sum_{i,j} f_{i,j} \max(0, 1-|x-x_i|) \max(0, 1-|y-y_i|)$$

$$f_{6.5,5.8} = (f_{6,5} * 0.5 * 0.2) + (f_{7,5} * 0.5 * 0.2) + (f_{6,6} * 0.5 * 0.8) + (f_{7,6} * 0.5 * 0.8)$$

He et al, “Mask R-CNN”, ICCV 2017

Cropping Features: RoI Align



Project proposal
onto features

No “snapping”!

Sample features at regularly-spaced points in each subregion using **bilinear interpolation**

After sampling, max-pool in each subregion

Input Image
(e.g. 3 x 640 x 480)

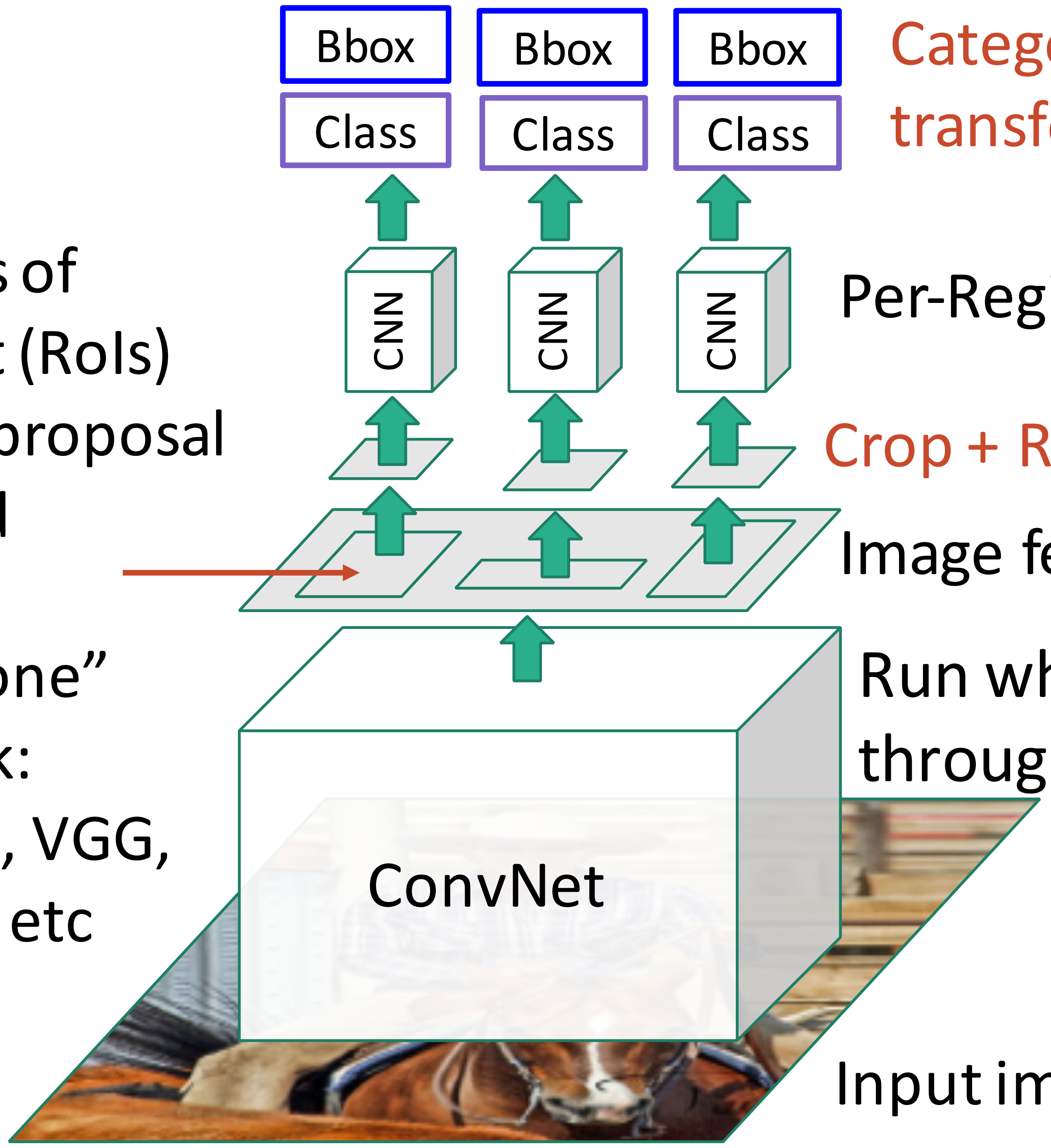
Image features
(e.g. 512 x 20 x 15)

Region features
(here 512 x 2 x 2;
In practice e.g 512 x 7 x 7)

He et al, “Mask R-CNN”, ICCV 2017

Fast R-CNN vs “Slow” R-CNN

Fast R-CNN: Apply differentiable cropping to shared image features



Category and box transform per region

Per-Region Network

Crop + Resize features

Image features

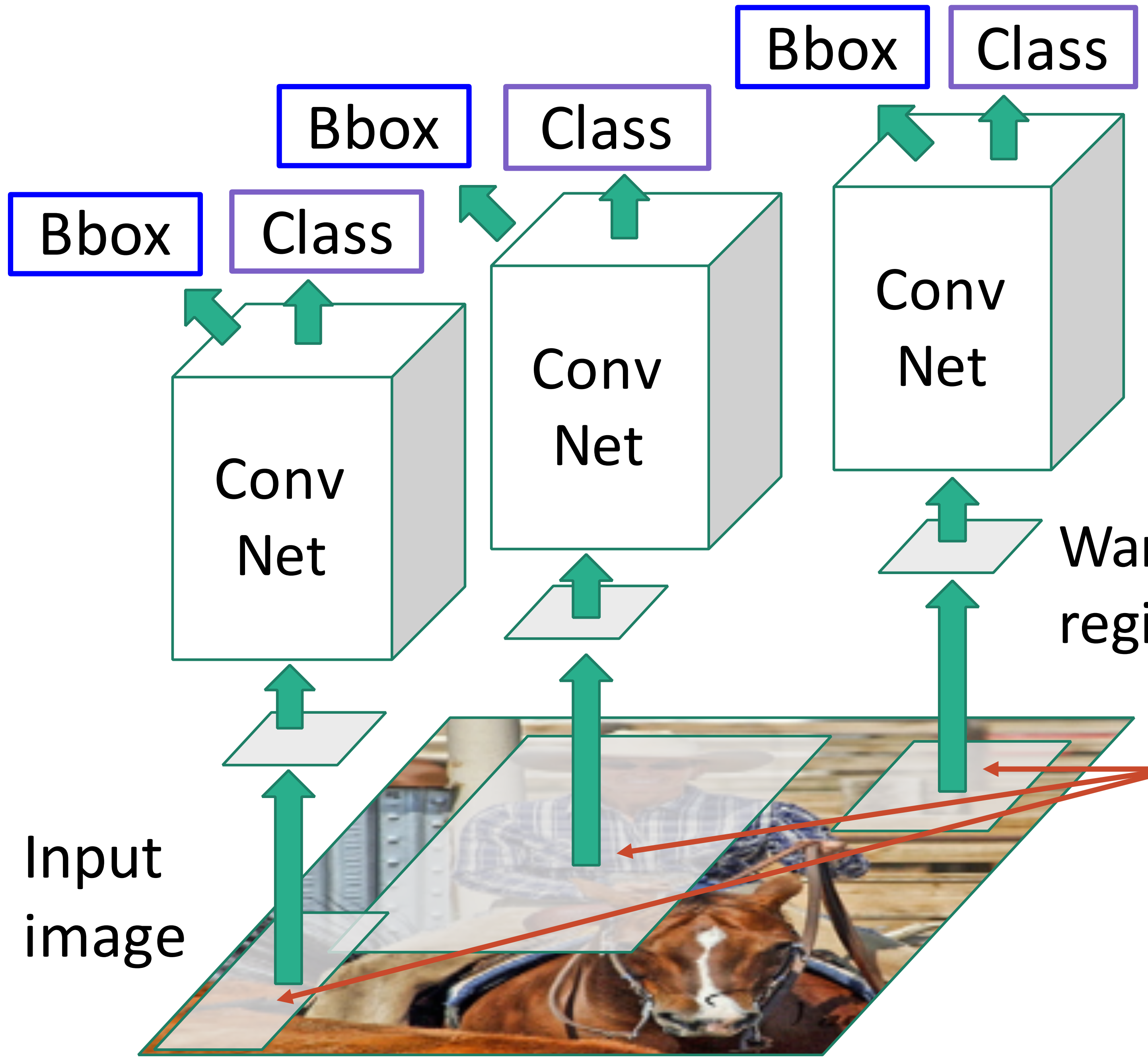
Regions of Interest (RoIs) from a proposal method

Run whole image through ConvNet

“Backbone” network: AlexNet, VGG, ResNet, etc

Input image

“Slow” R-CNN: Apply differentiable cropping to shared image features



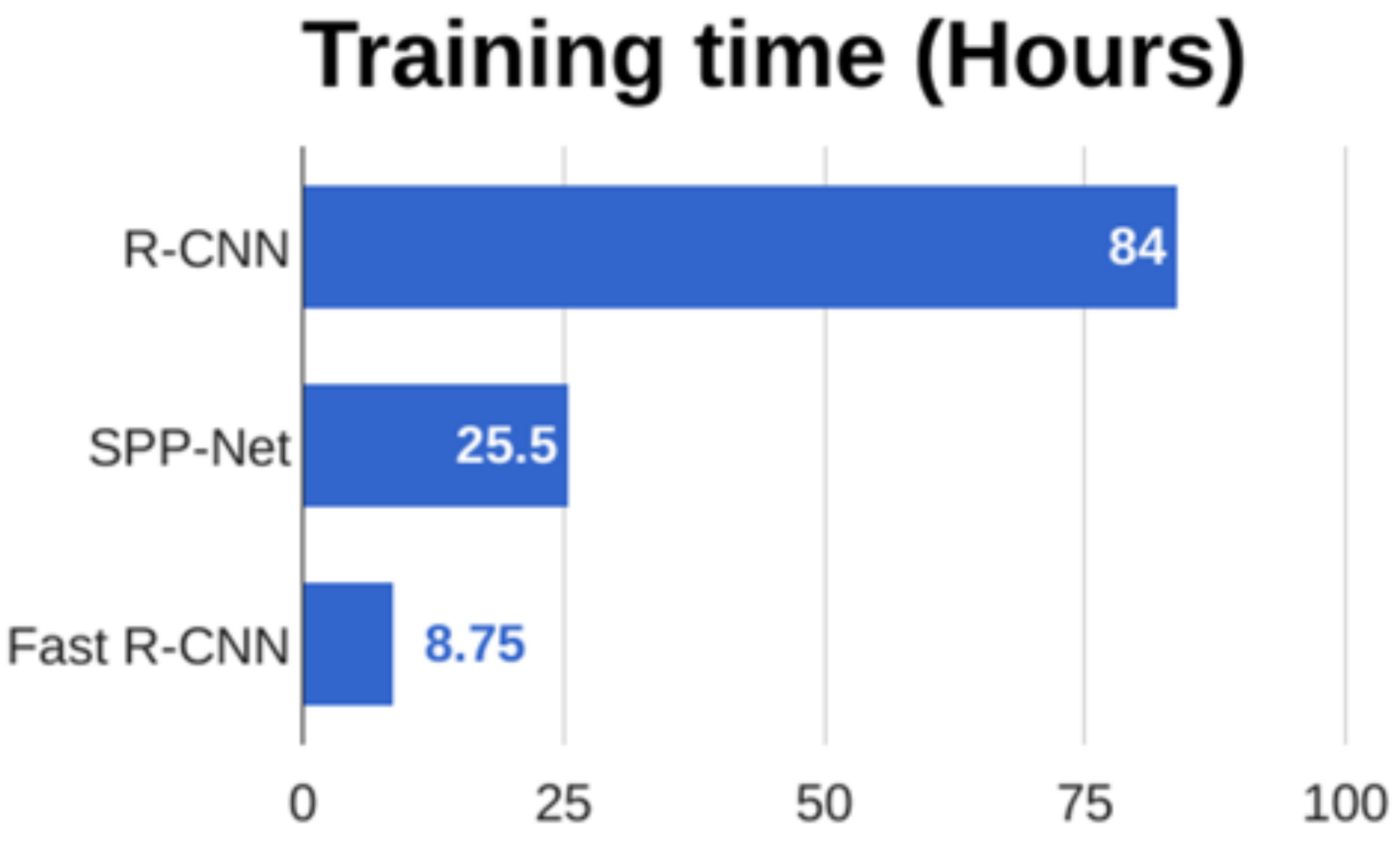
Forward each region through ConvNet

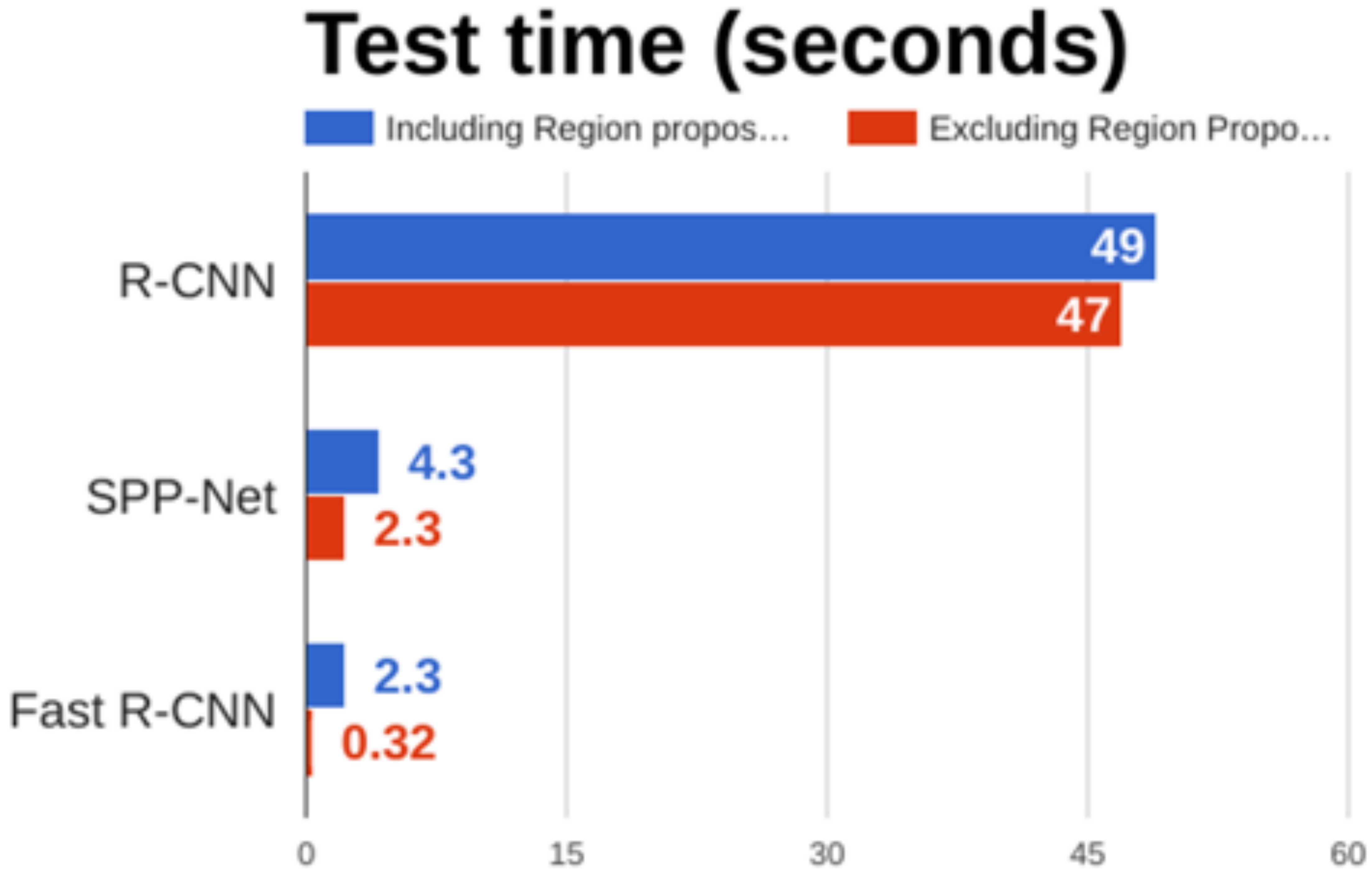
Warped image regions (224x224)

Input image

Regions of Interest (RoI) from a proposal method (~2k)

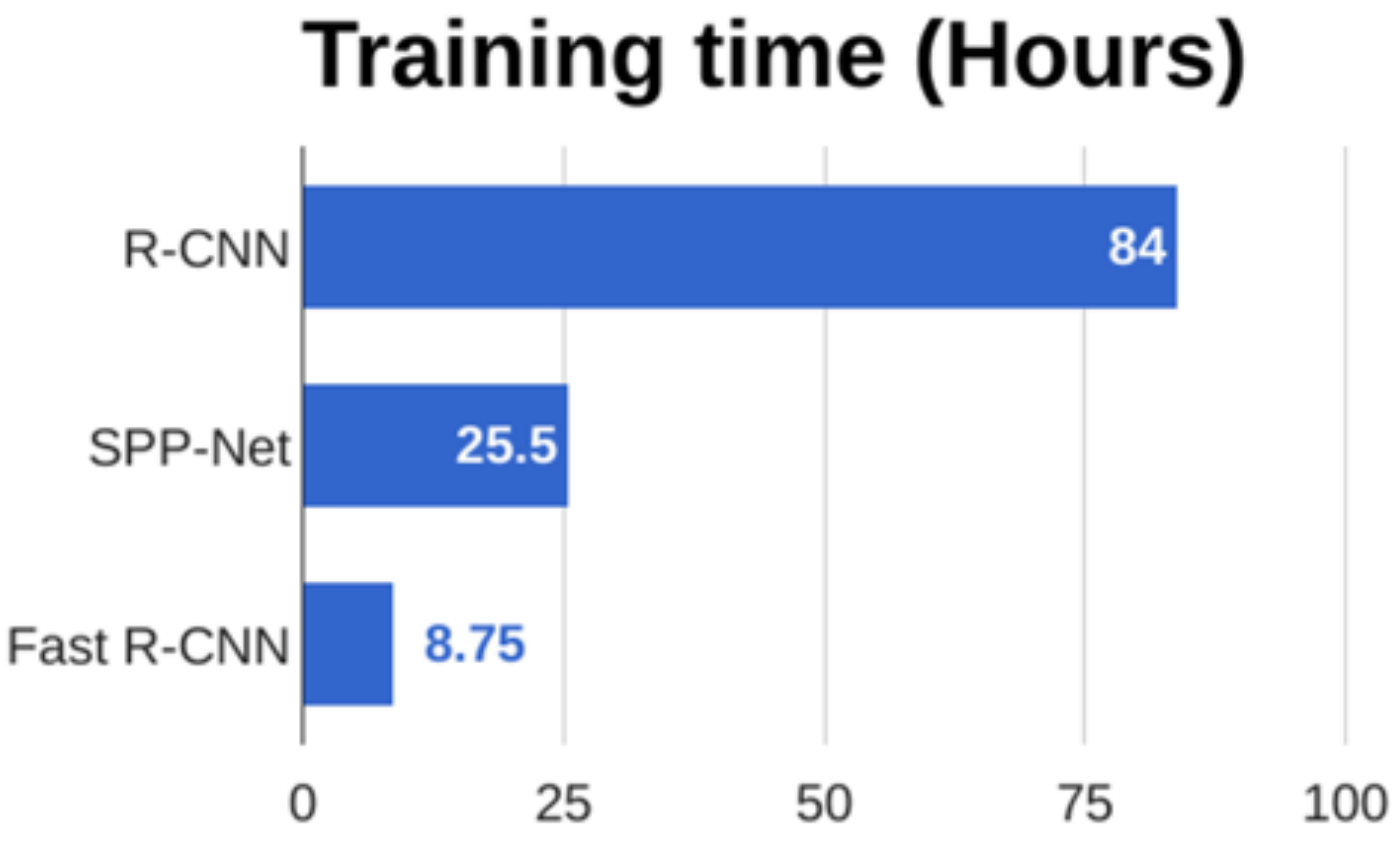
Fast R-CNN vs "Slow" R-CNN



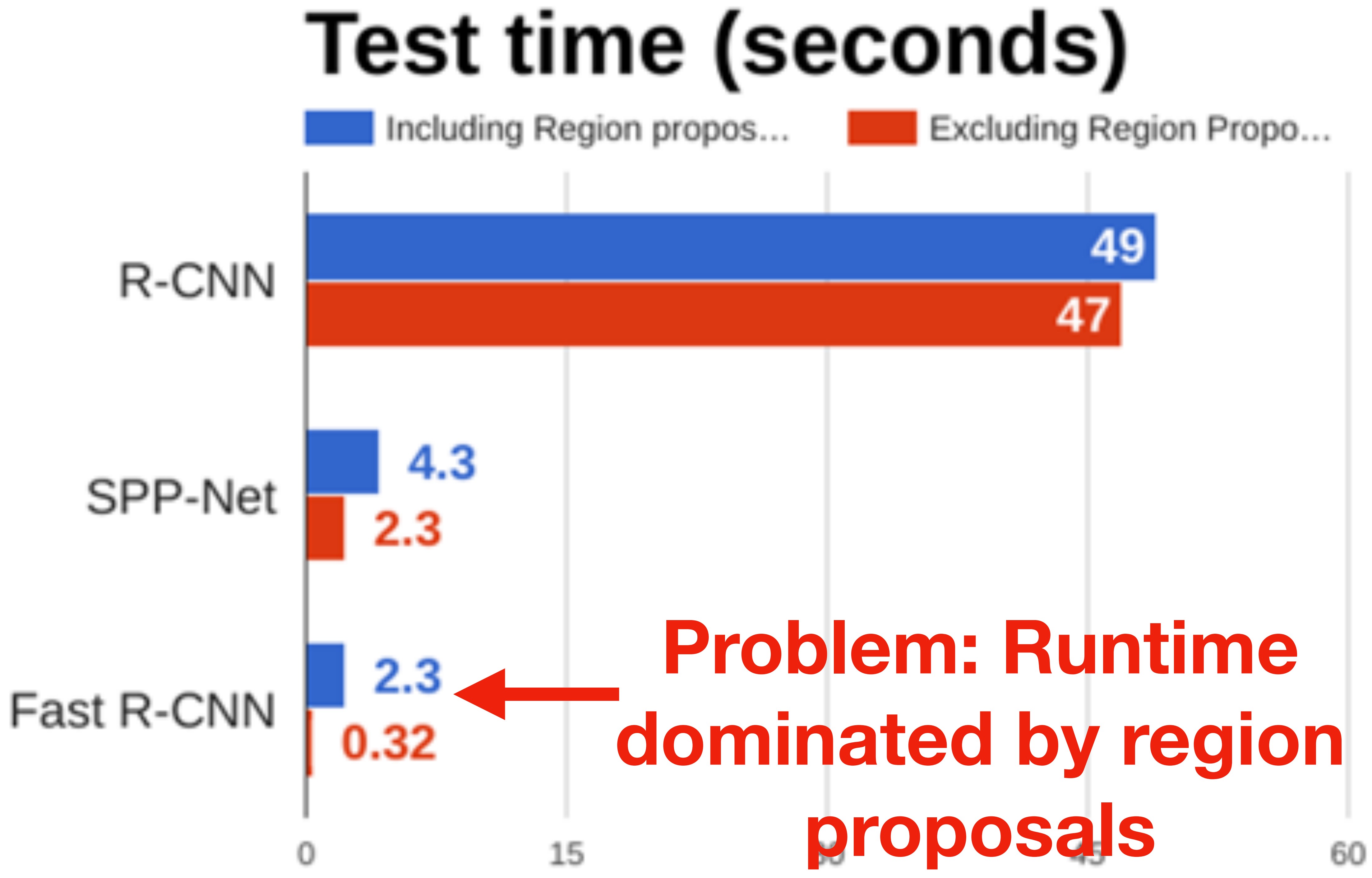


Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
He et al, "Spatial pyramid pooling in deep convolutional networks for visual recognition", ECCV 2014
Girshick, "Fast R-CNN", ICCV 2015

Fast R-CNN vs "Slow" R-CNN



Training time (Hours)

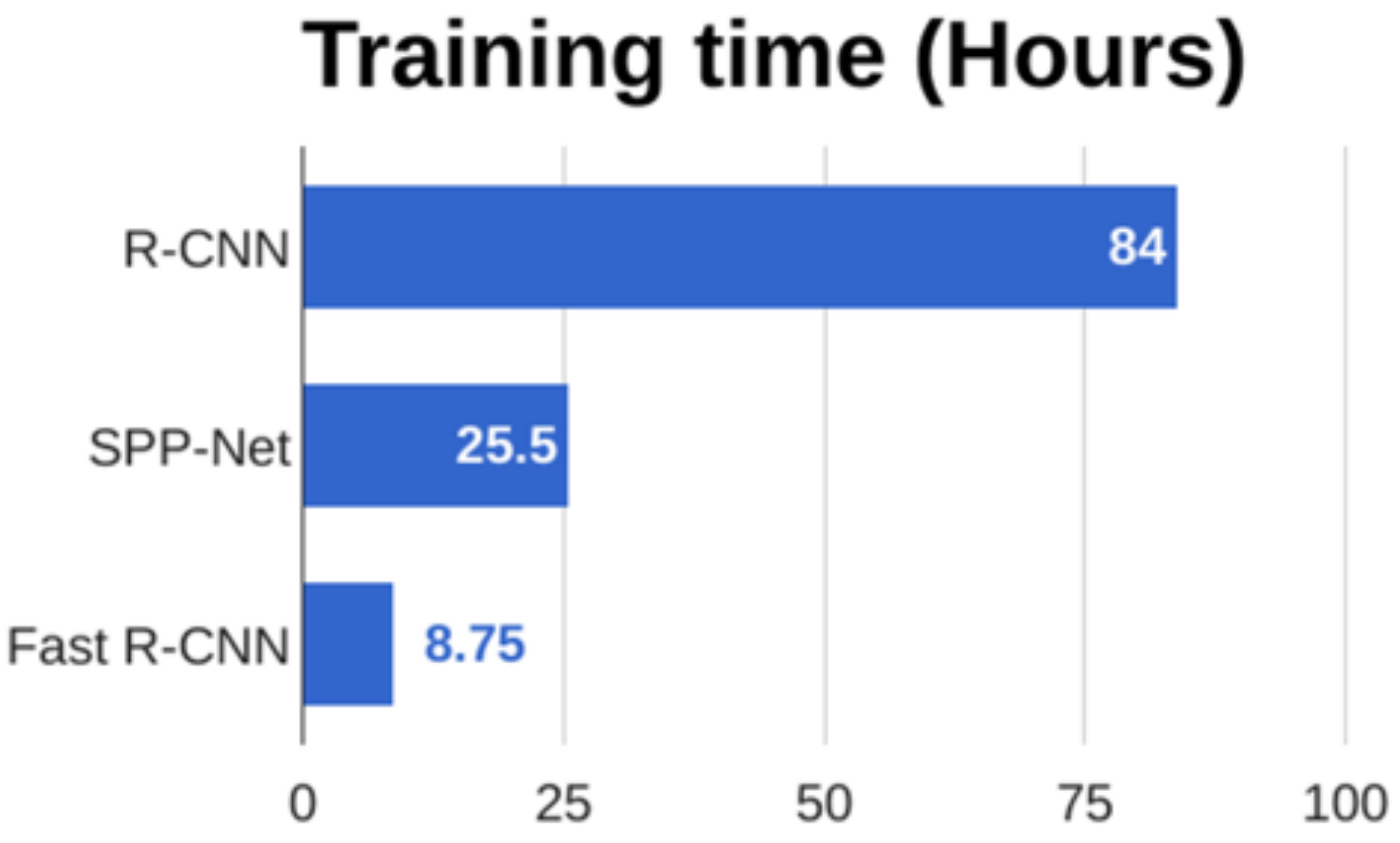


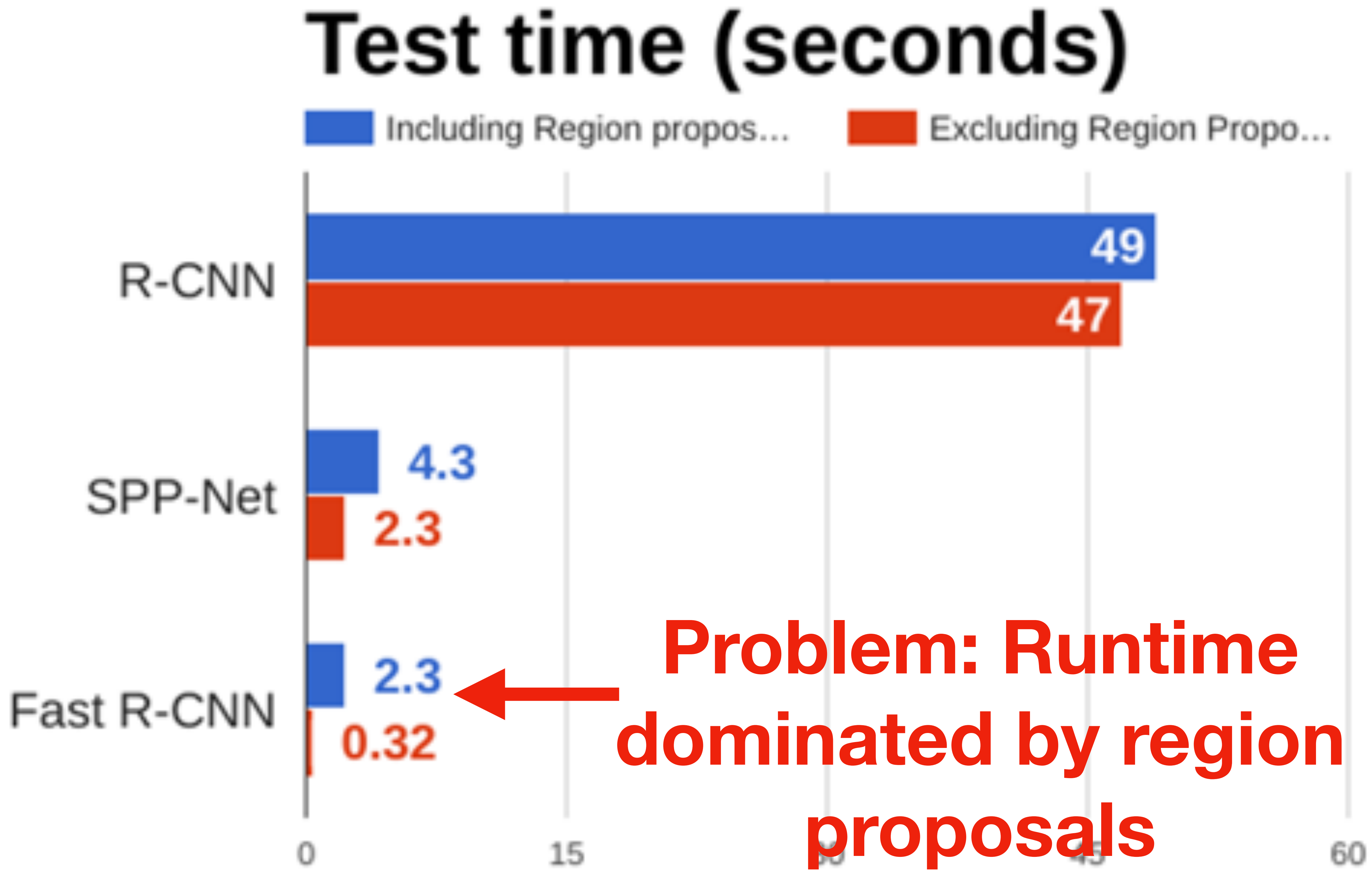
Test time (seconds)

Problem: Runtime dominated by region proposals

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
He et al, "Spatial pyramid pooling in deep convolutional networks for visual recognition", ECCV 2014
Girshick, "Fast R-CNN", ICCV 2015

Fast R-CNN vs “Slow” R-CNN





Problem: Runtime dominated by region proposals

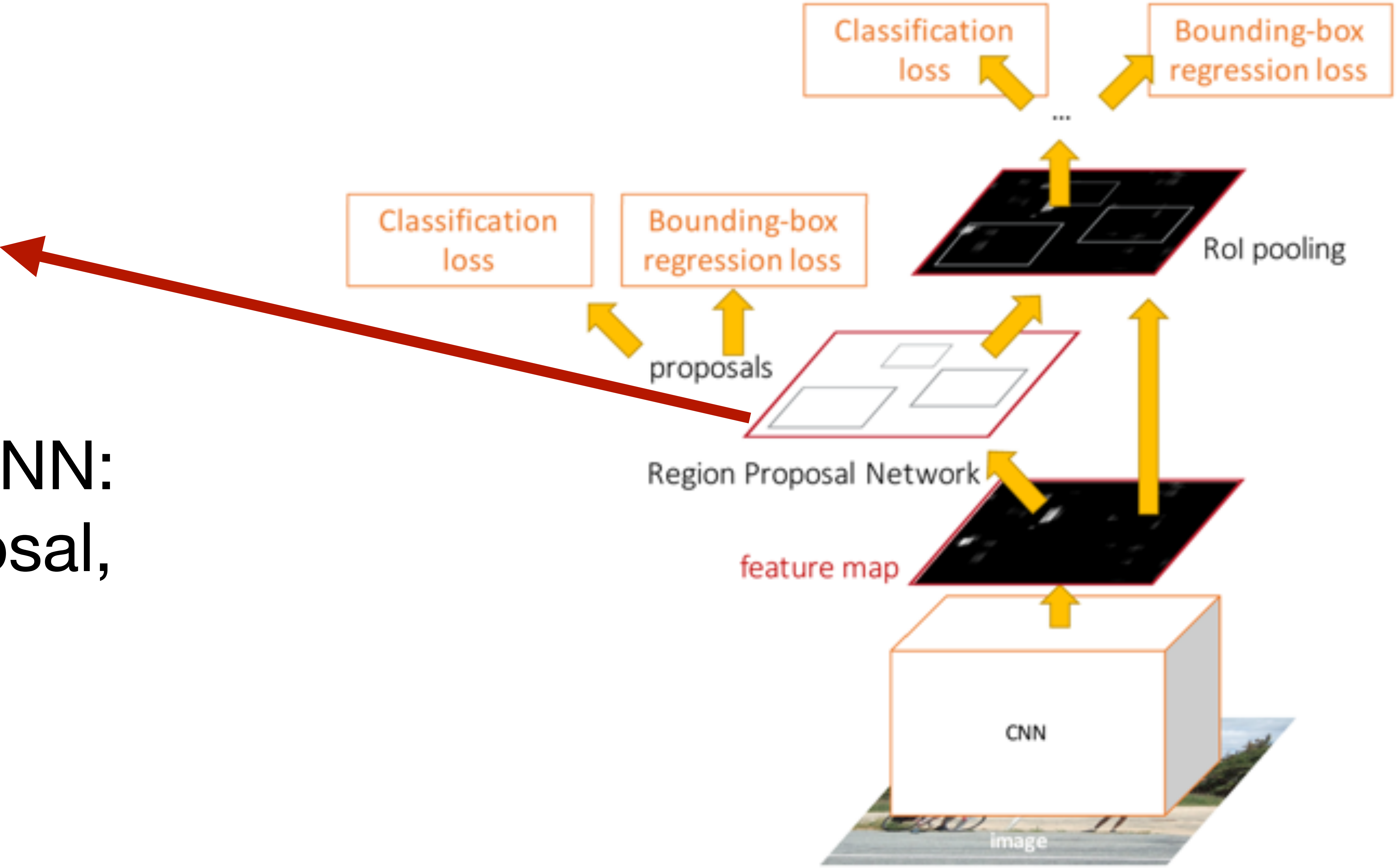
Recall: Region proposals computed by heuristic “Selective search” algorithm on CPU — let’s learn them with a CNN

Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
He et al, “Spatial pyramid pooling in deep convolutional networks for visual recognition”, ECCV 2014
Girshick, “Fast R-CNN”, ICCV 2015

Faster R-CNN: Learnable Region Proposals

Insert **Region Proposal Network (RPN)** to predict proposals from features

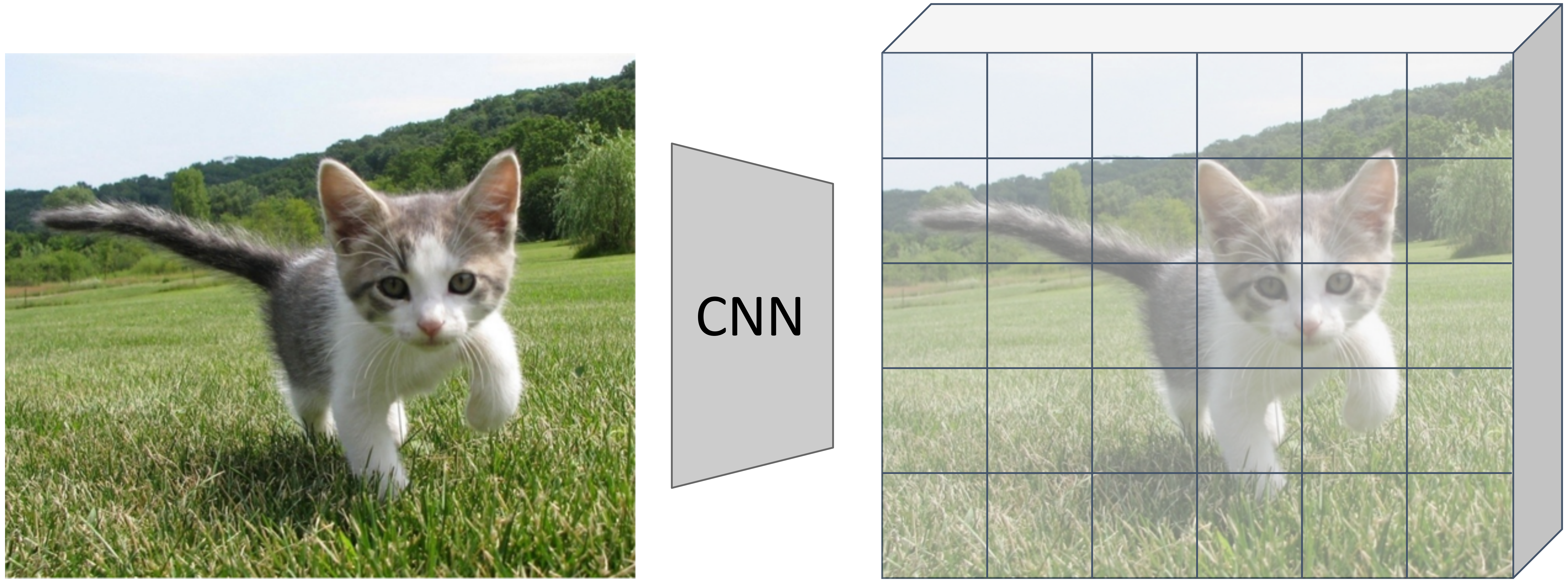
Otherwise same as Fast R-CNN: Crop features for each proposal, classify each one



Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

Region Proposal Network (RPN)

Run backbone CNN to get
features aligned to input image



Input Image
(e.g. 3 x 640 x 480)

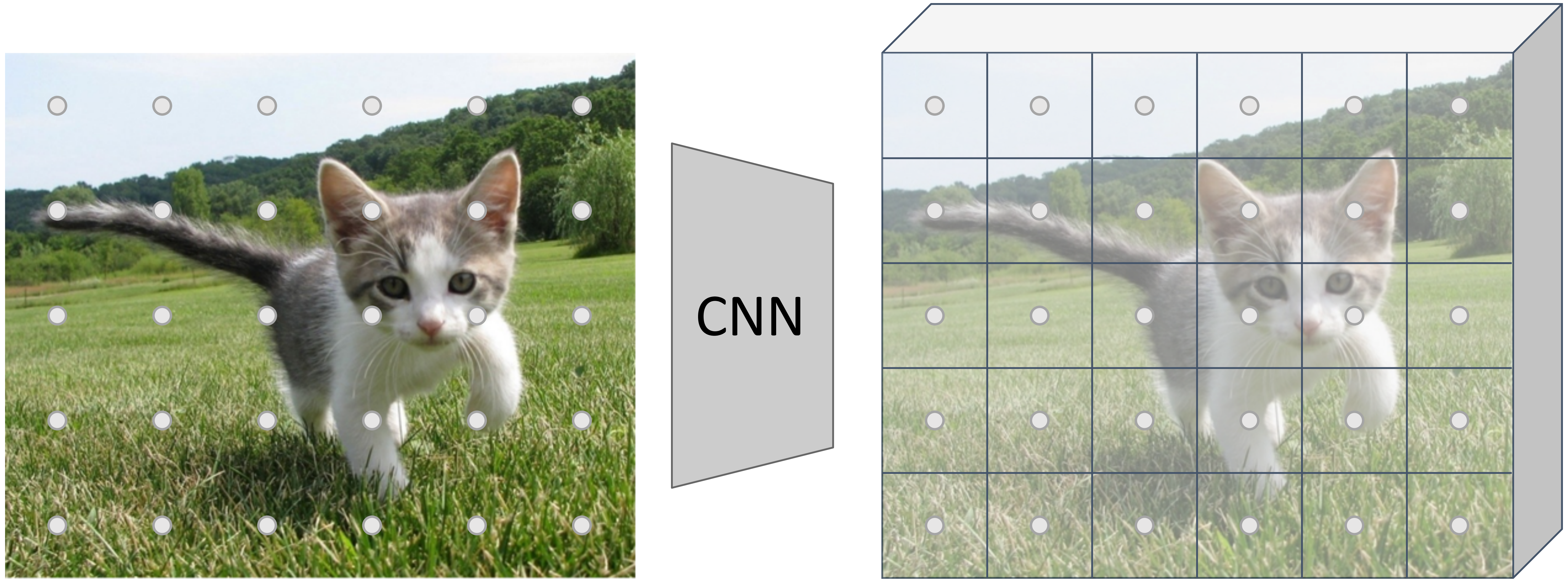
Image features
(e.g. 512 x 5 x 6)

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

Region Proposal Network (RPN)

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input



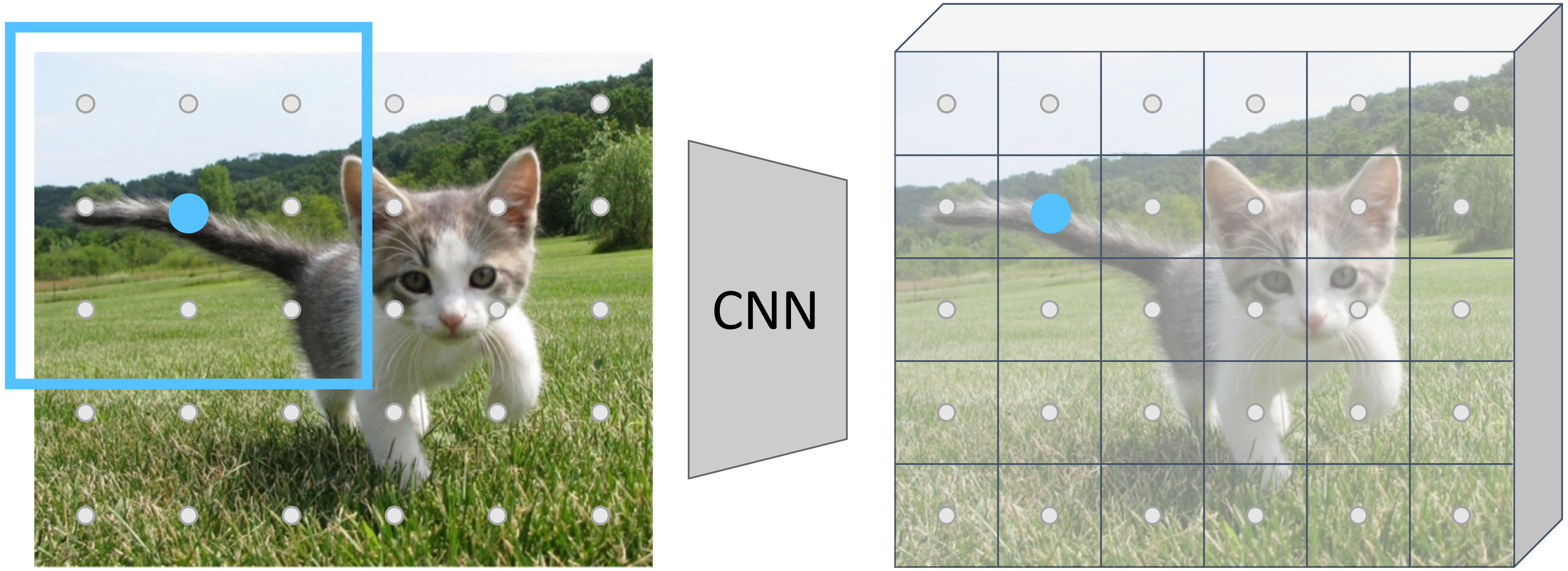
Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

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

Region Proposal Network (RPN)

Imagine an **anchor box** of fixed size at each point in the feature map



Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

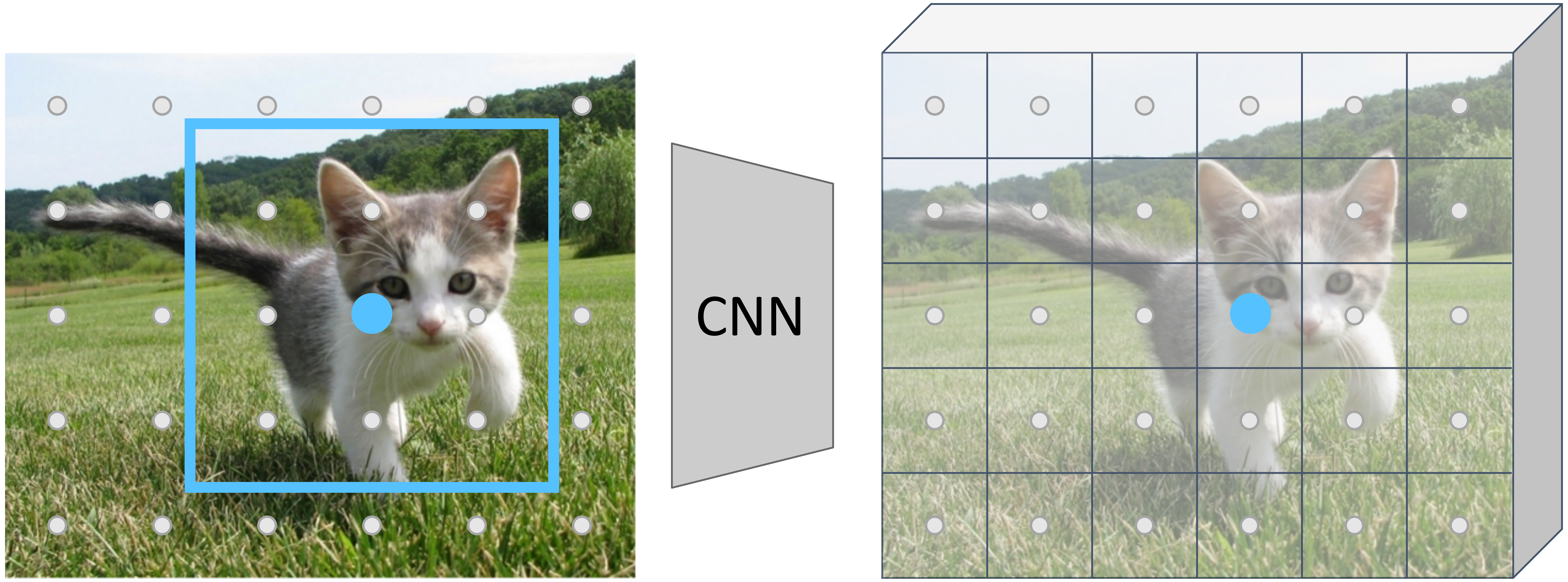
Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

Region Proposal Network (RPN)

Imagine an **anchor box** of fixed size at each point in the feature map

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input



Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

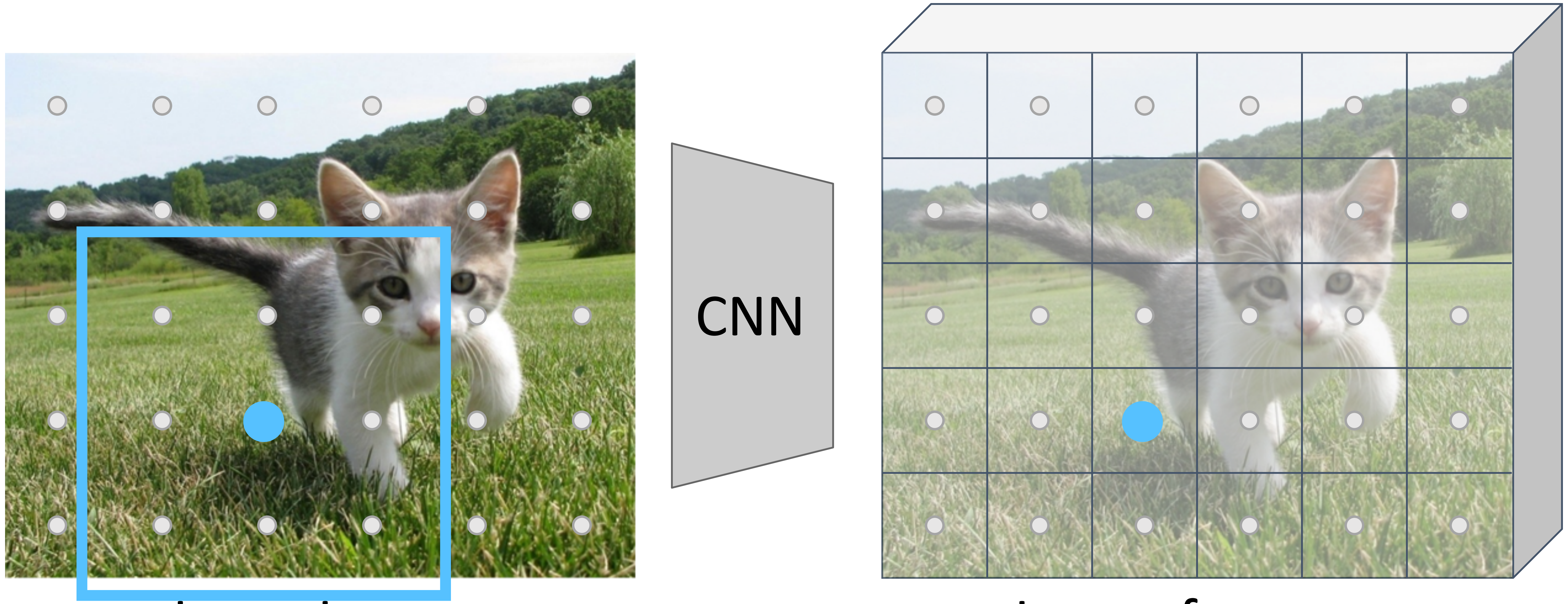
Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

Region Proposal Network (RPN)

Imagine an **anchor box** of fixed size at each point in the feature map

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input



Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

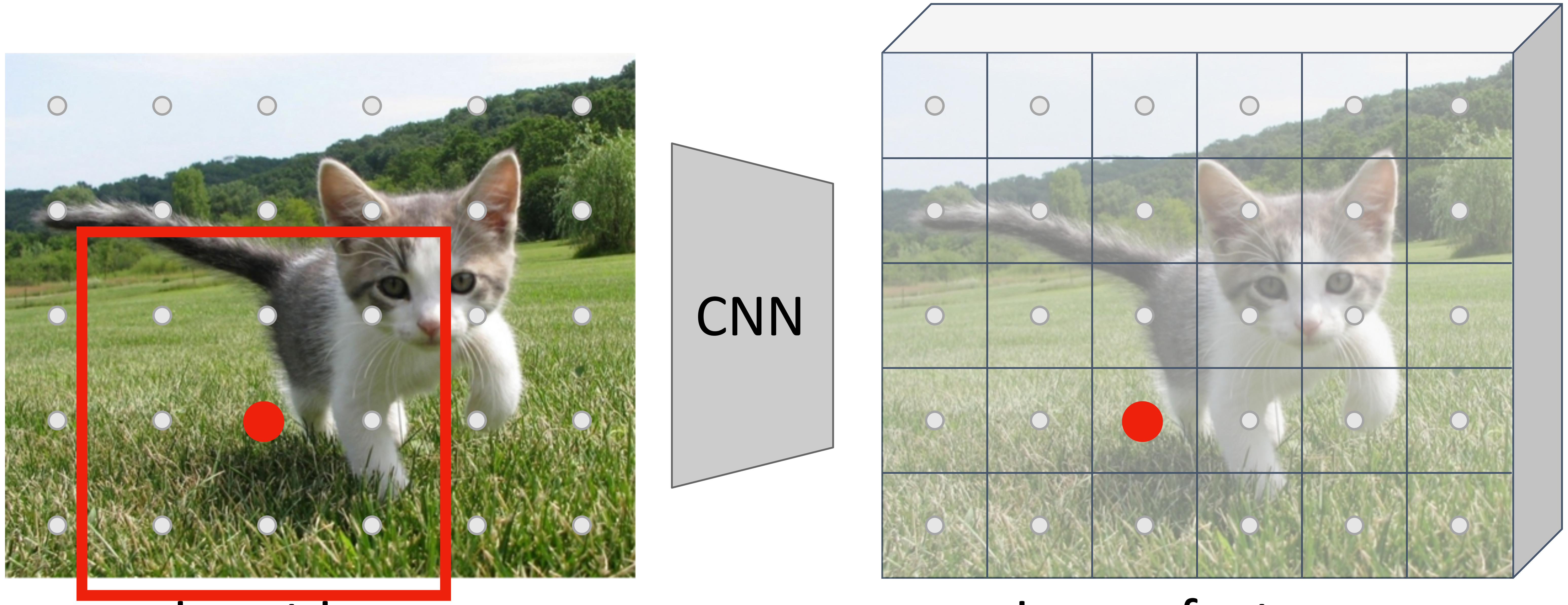
Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Region Proposal Network (RPN)

Imagine an **anchor box** of fixed size at each point in the feature map

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input



Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Classify each anchor as positive (object) or negative (no object)

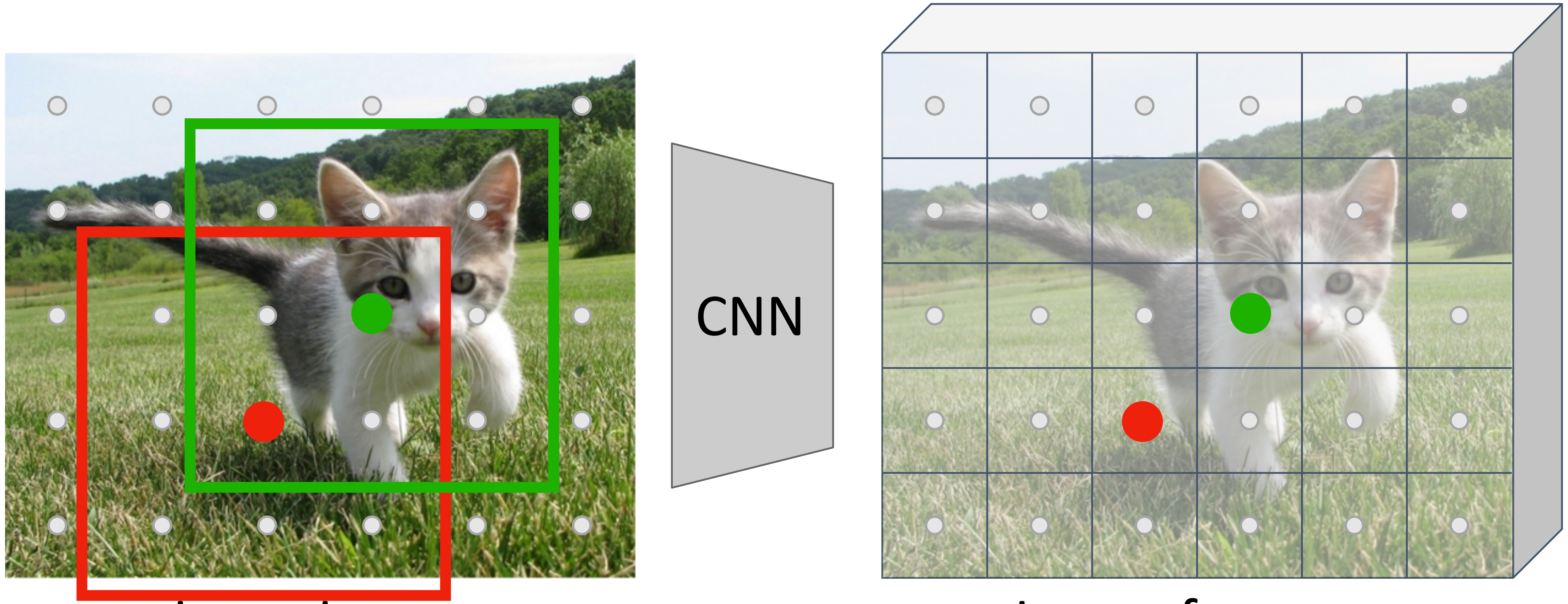
Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

Region Proposal Network (RPN)

Imagine an **anchor box** of fixed size at each point in the feature map

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input



Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Classify each anchor as positive (object) or negative (no object)

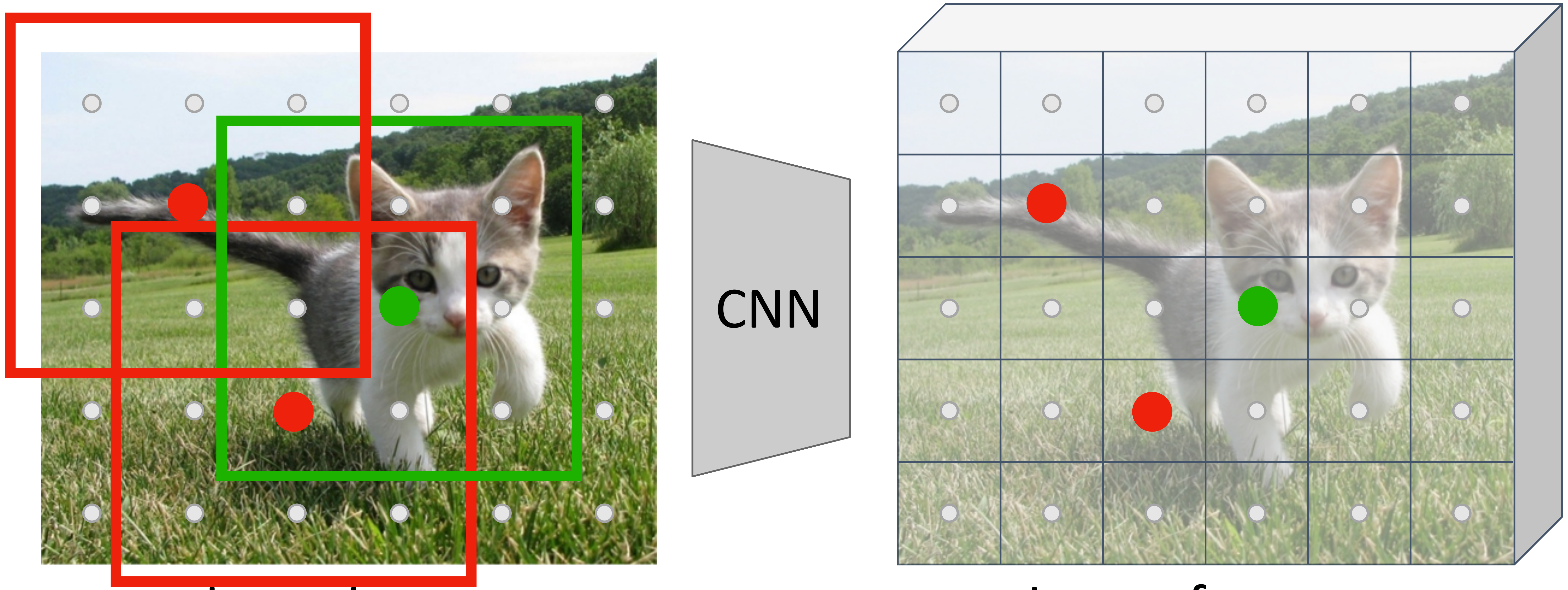
Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Region Proposal Network (RPN)

Imagine an **anchor box** of fixed size at each point in the feature map

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input



Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Classify each anchor as positive (object) or negative (no object)

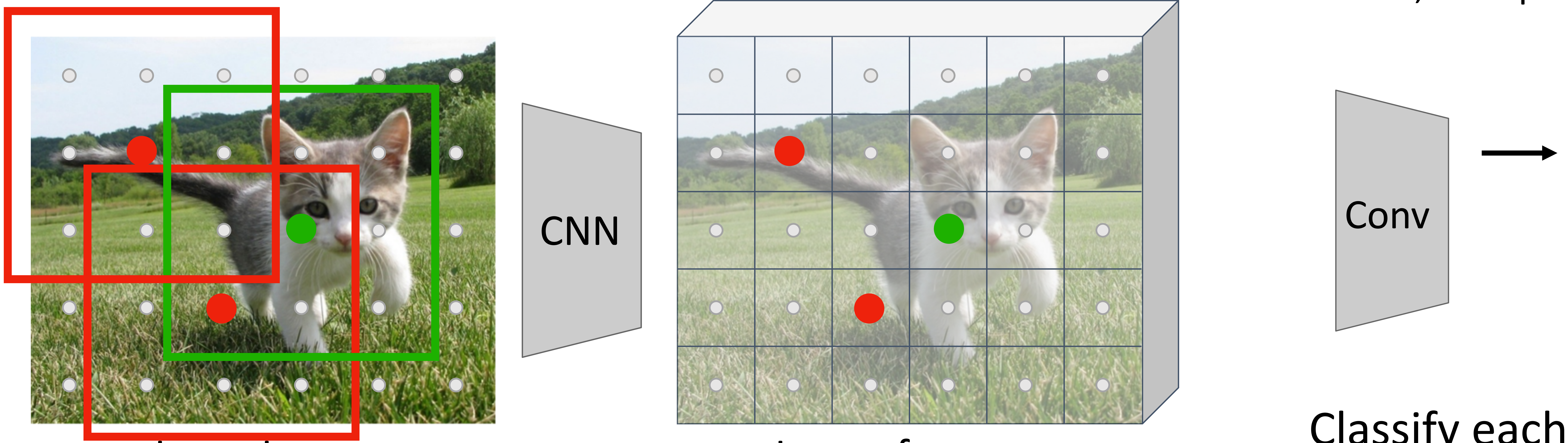
Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Region Proposal Network (RPN)

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

Predict object vs not object scores for all anchors with a conv layer (512 input filters, 2 output filters)



Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Anchor is object?
2 x 5 x 6

Classify each anchor as positive (object) or negative (no object)

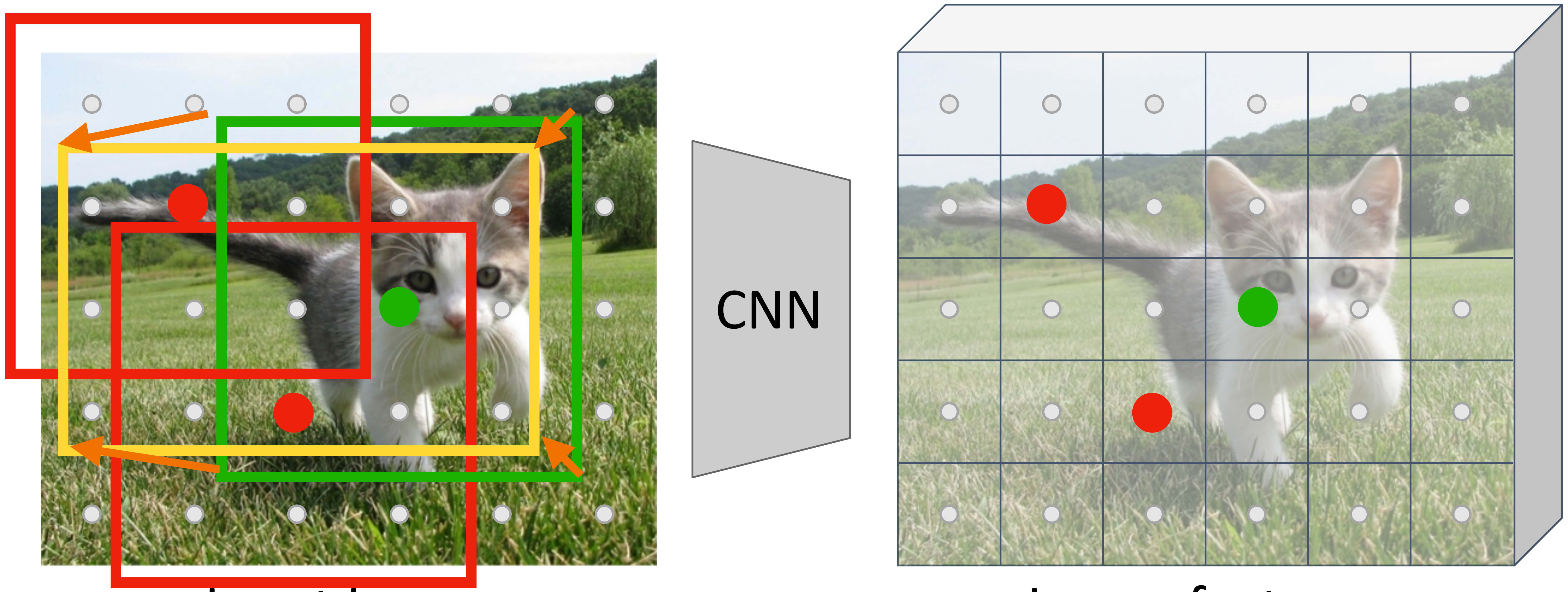
Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Region Proposal Network (RPN)

For positive anchors, also predict a transform that converting the anchor to the GT box (like R-CNN)

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input



Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Conv → Anchor is object?
2 x 5 x 6

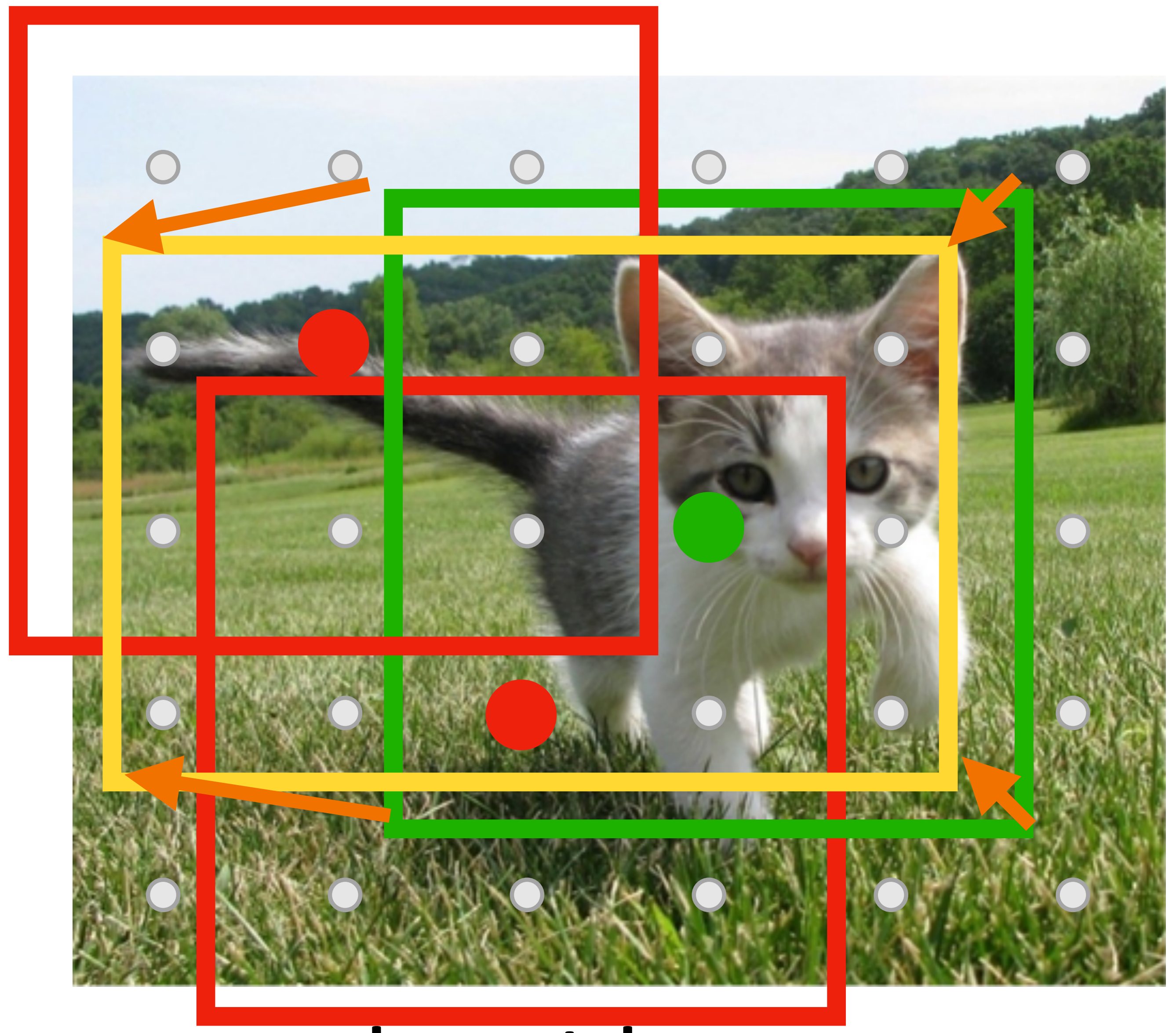
Classify each anchor as positive (object) or negative (no object)

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

Region Proposal Network (RPN)

For positive anchors, also predict a transform that converting the anchor to the GT box (like R-CNN)

Run backbone CNN to get features aligned to input image



Input Image
(e.g. 3 x 640 x 480)

CNN

Each feature corresponds to a point in the input

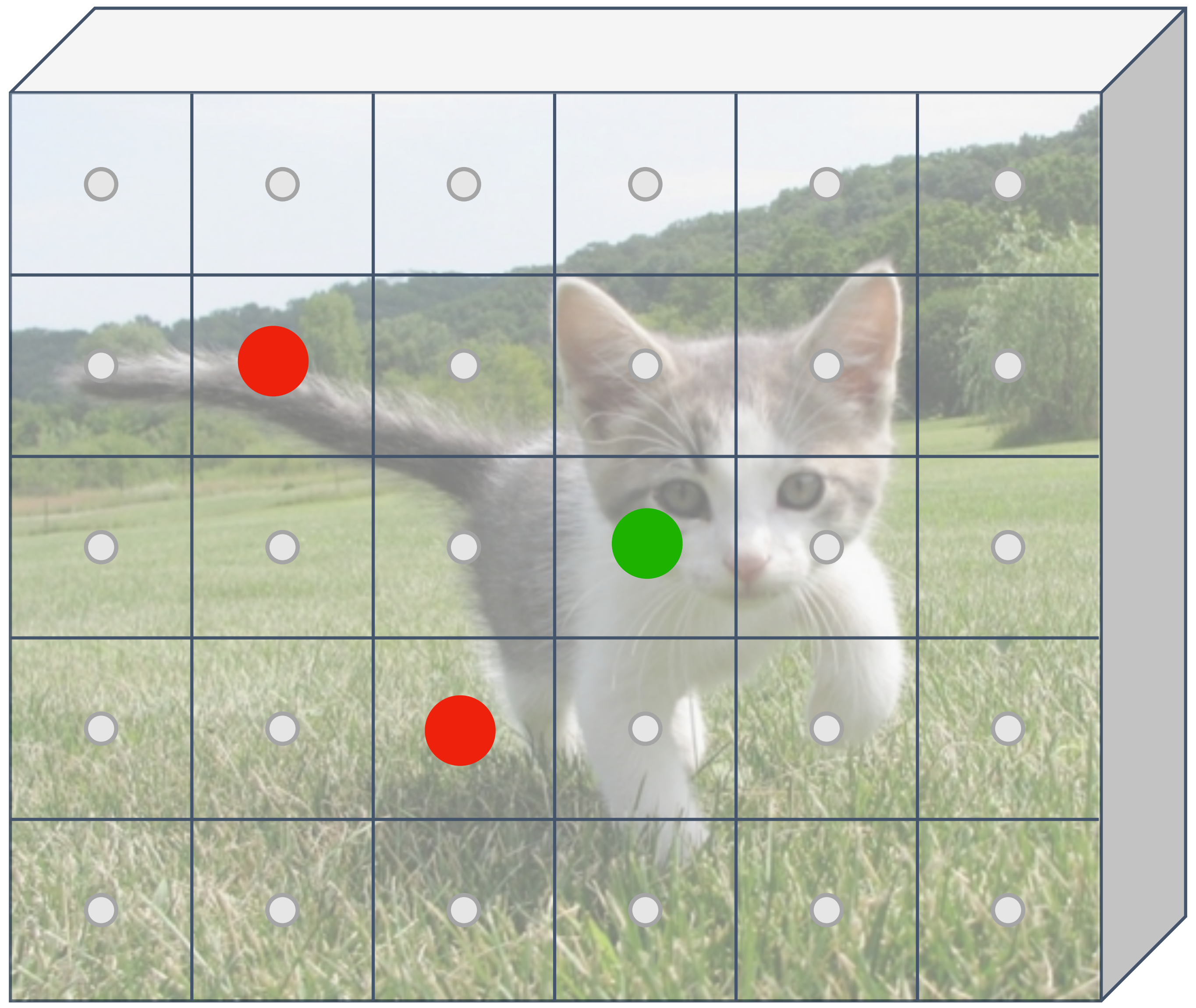


Image features
(e.g. 512 x 5 x 6)

Conv

→ Anchor is object?
2 x 5 x 6

→ Anchor transforms
4 x 5 x 6

Classify each anchor as positive (object) or negative (no object)

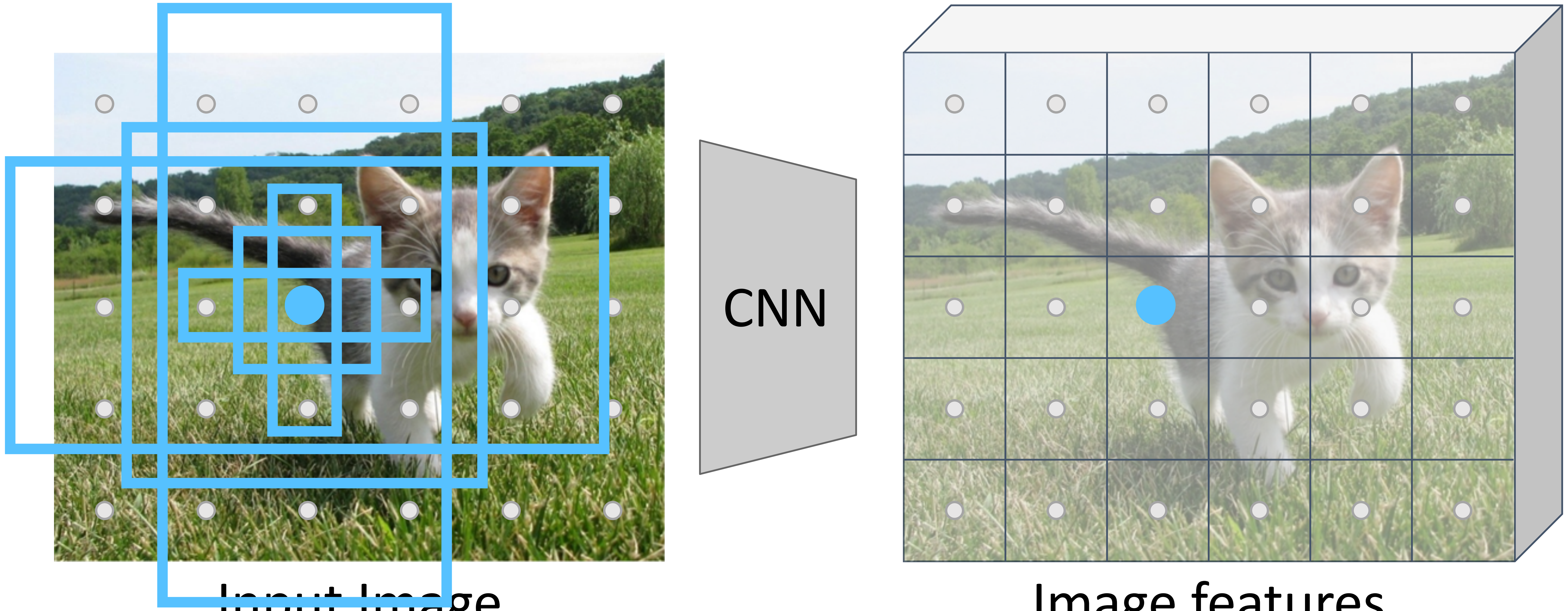
Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

Region Proposal Network (RPN)

Run backbone CNN to get features aligned to input image

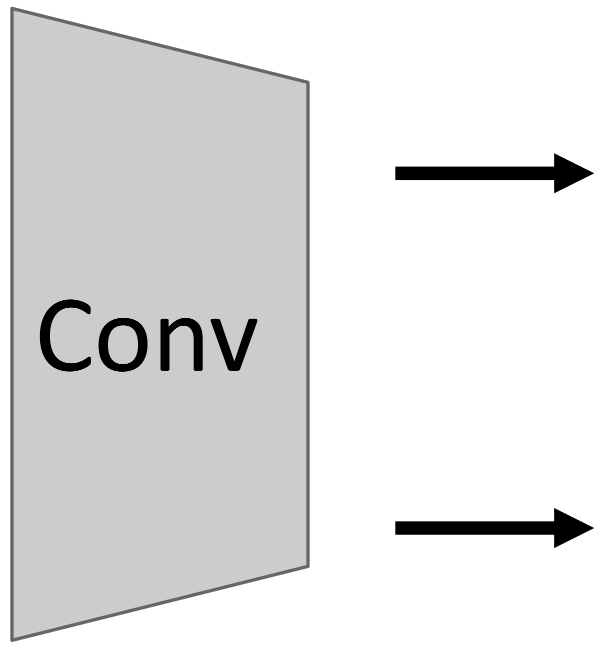
Each feature corresponds to a point in the input

In practice: Rather than using one anchor per point, instead consider K different anchors with different size and scale (here K = 6)



Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)



Anchor is object?
2K x 5 x 6

Anchor transforms
4K x 5 x 6

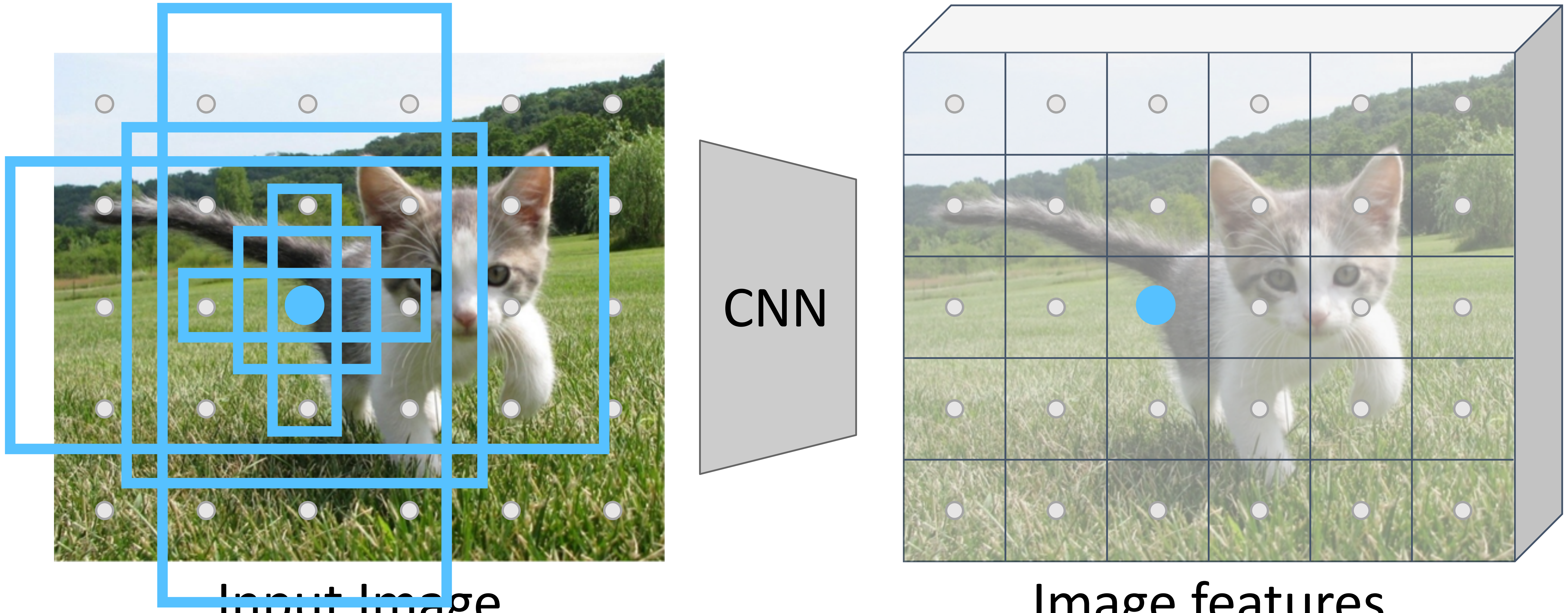
Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Region Proposal Network (RPN)

Run backbone CNN to get features aligned to input image

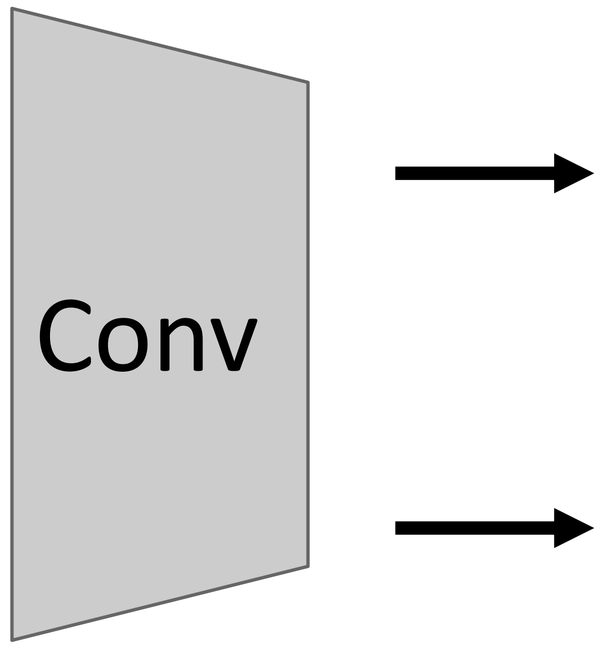
Each feature corresponds to a point in the input

In practice: Rather than using one anchor per point, instead consider K different anchors with different size and scale (here K = 6)



Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)



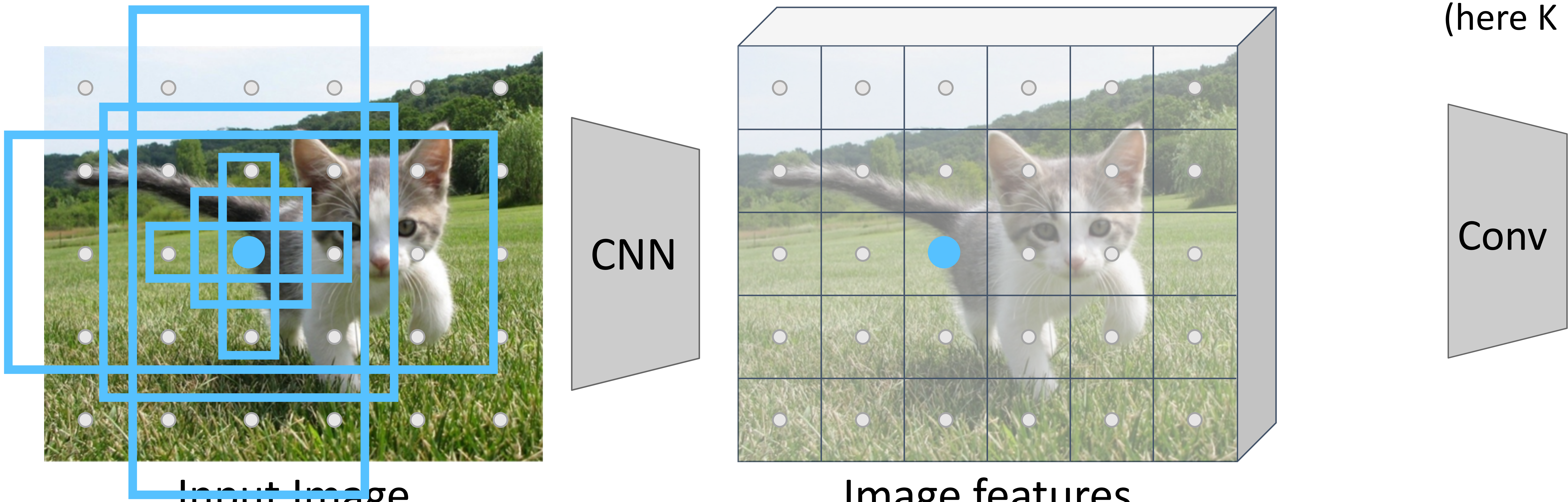
Anchor is object?
2K x 5 x 6

Anchor transforms
4K x 5 x 6

During training, supervised positive / negative anchors and box transforms like R-CNN

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

Region Proposal Network (RPN)



Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input

In practice: Rather than using one anchor per point, instead consider K different anchors with different size and scale (here K = 6)

Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Positive anchors: >= 0.7 IoU with some GT box (plus highest IoU to each GT)

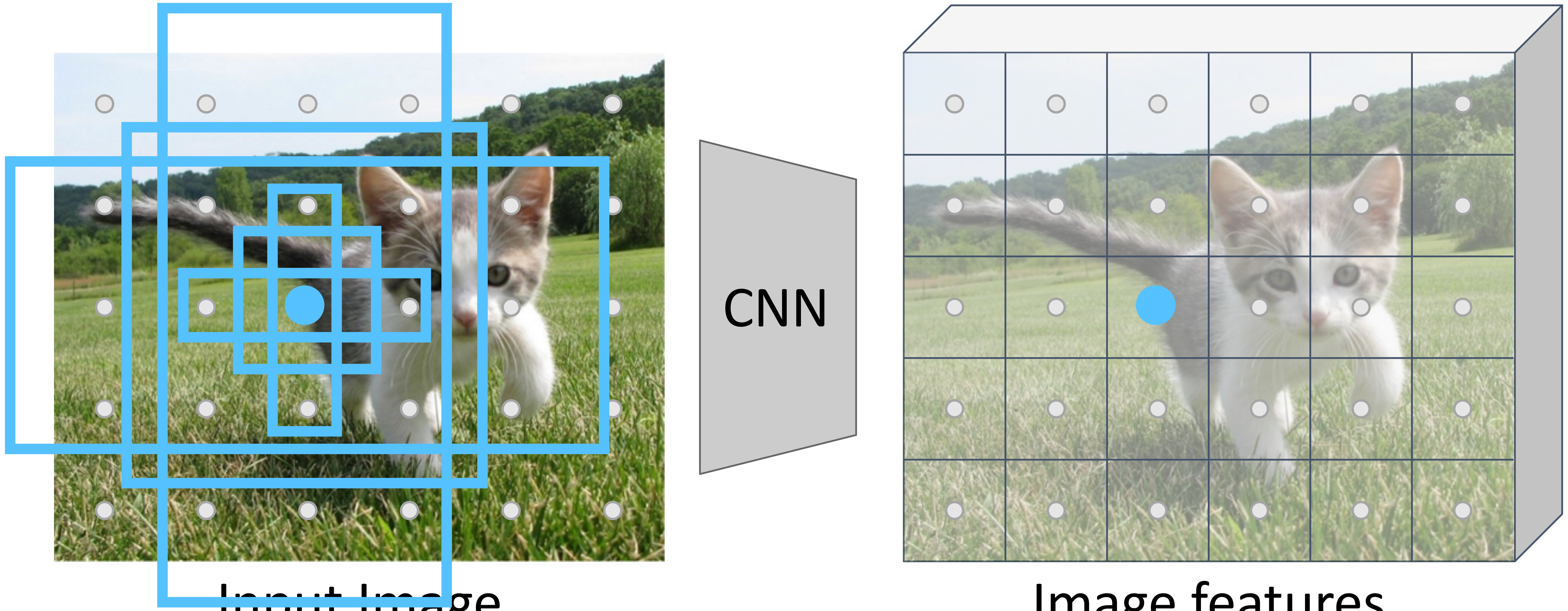
Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

Region Proposal Network (RPN)

Run backbone CNN to get features aligned to input image

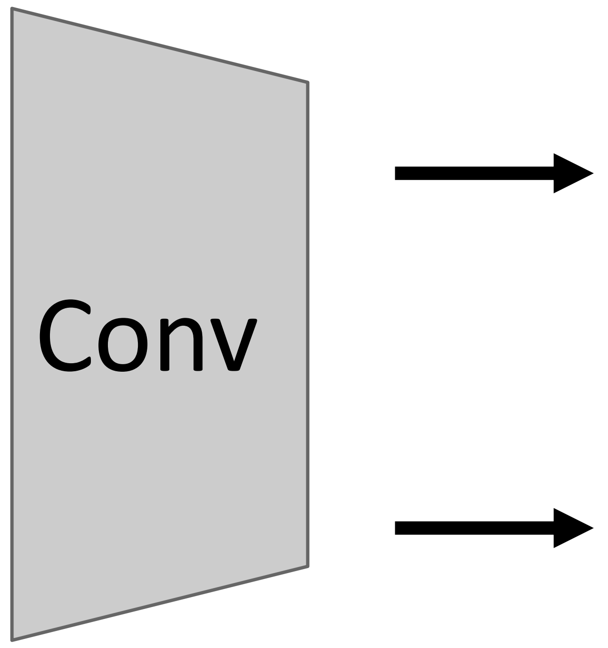
Each feature corresponds to a point in the input

In practice: Rather than using one anchor per point, instead consider K different anchors with different size and scale (here K = 6)



Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)



Anchor is object?
2K x 5 x 6

Anchor transforms
4K x 5 x 6

Negative anchors: < 0.3 IoU with all GT boxes. Don’t supervised transforms for negative boxes.

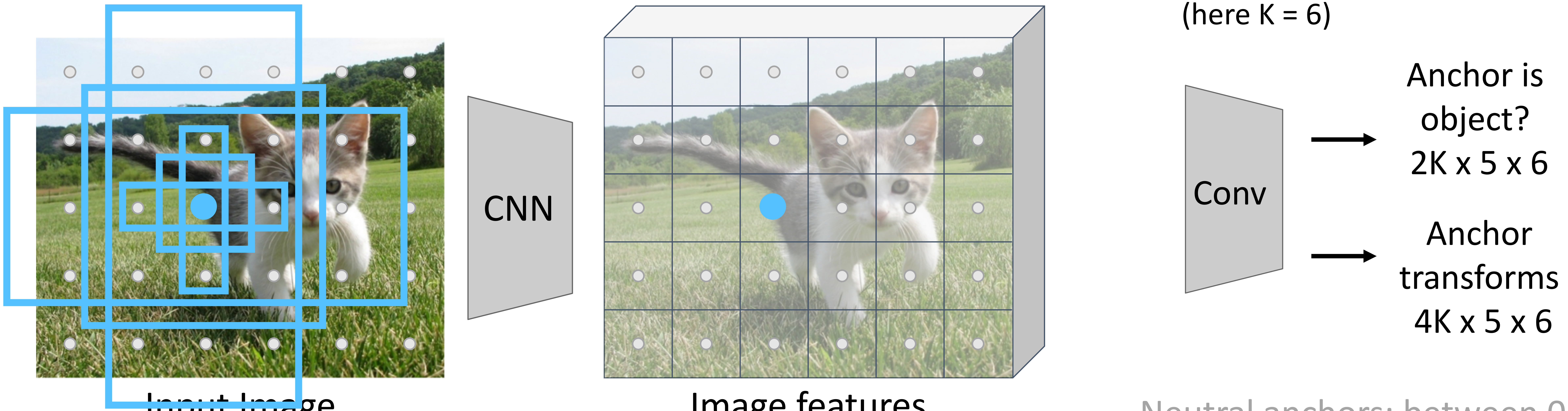
Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Region Proposal Network (RPN)

In practice: Rather than using one anchor per point, instead consider K different anchors with different size and scale (here K = 6)

Run backbone CNN to get features aligned to input image

Each feature corresponds to a point in the input



Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)

Neutral anchors: between 0.3 and 0.7 IoU with all GT boxes; ignored during training

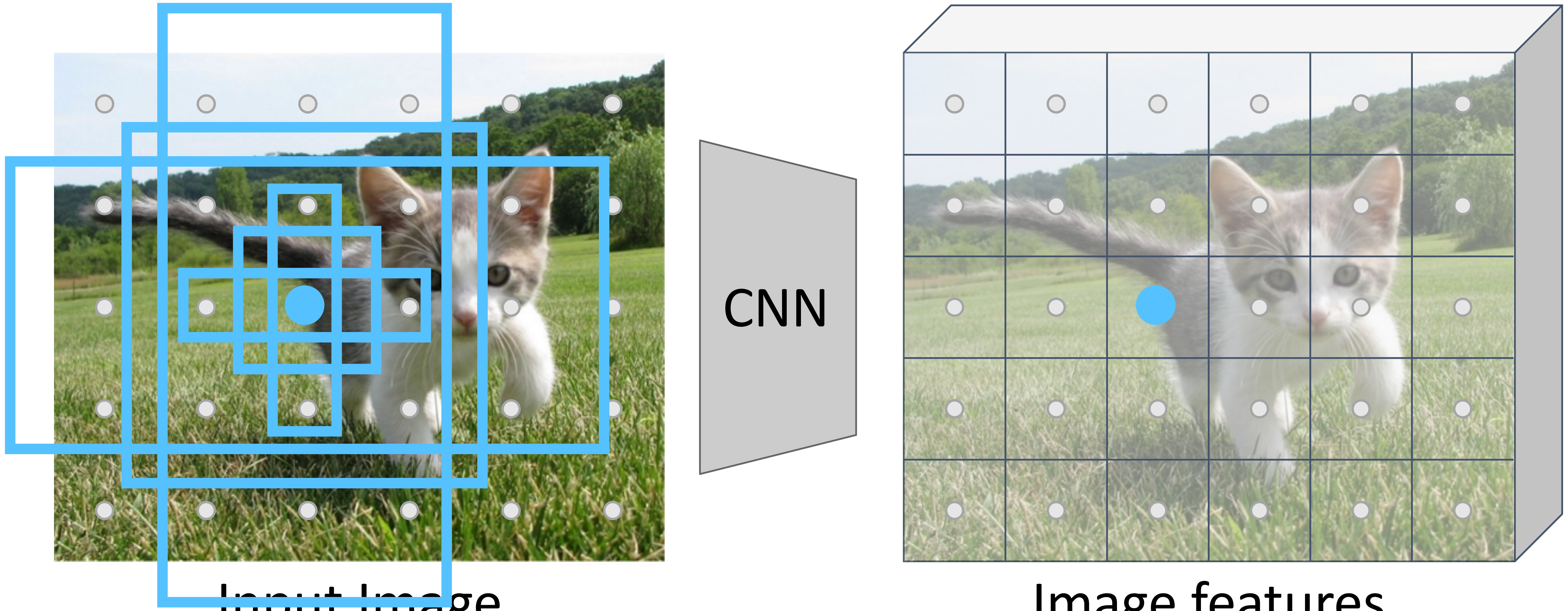
Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

Region Proposal Network (RPN)

Run backbone CNN to get features aligned to input image

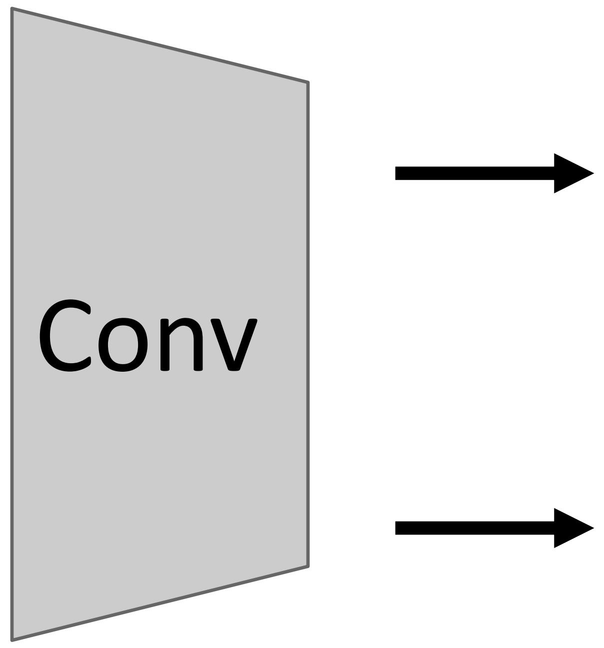
Each feature corresponds to a point in the input

In practice: Rather than using one anchor per point, instead consider K different anchors with different size and scale (here K = 6)



Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 5 x 6)



Anchor is object?
2K x 5 x 6

Anchor transforms
4K x 5 x 6

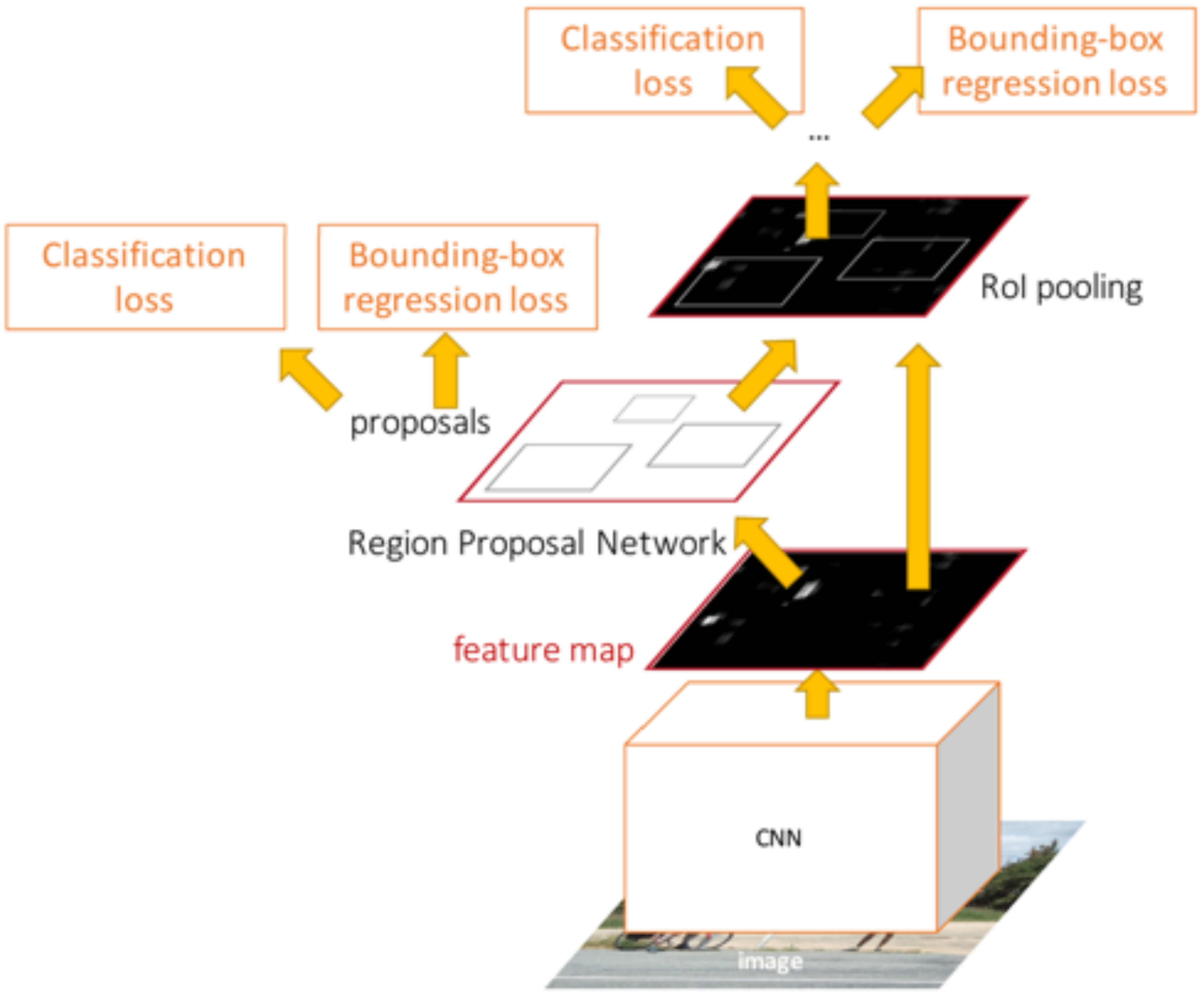
At test-time, sort all K*5*6 boxes by their positive score, take top 300 as our region proposals

Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

Faster R-CNN: Learnable Region Proposals

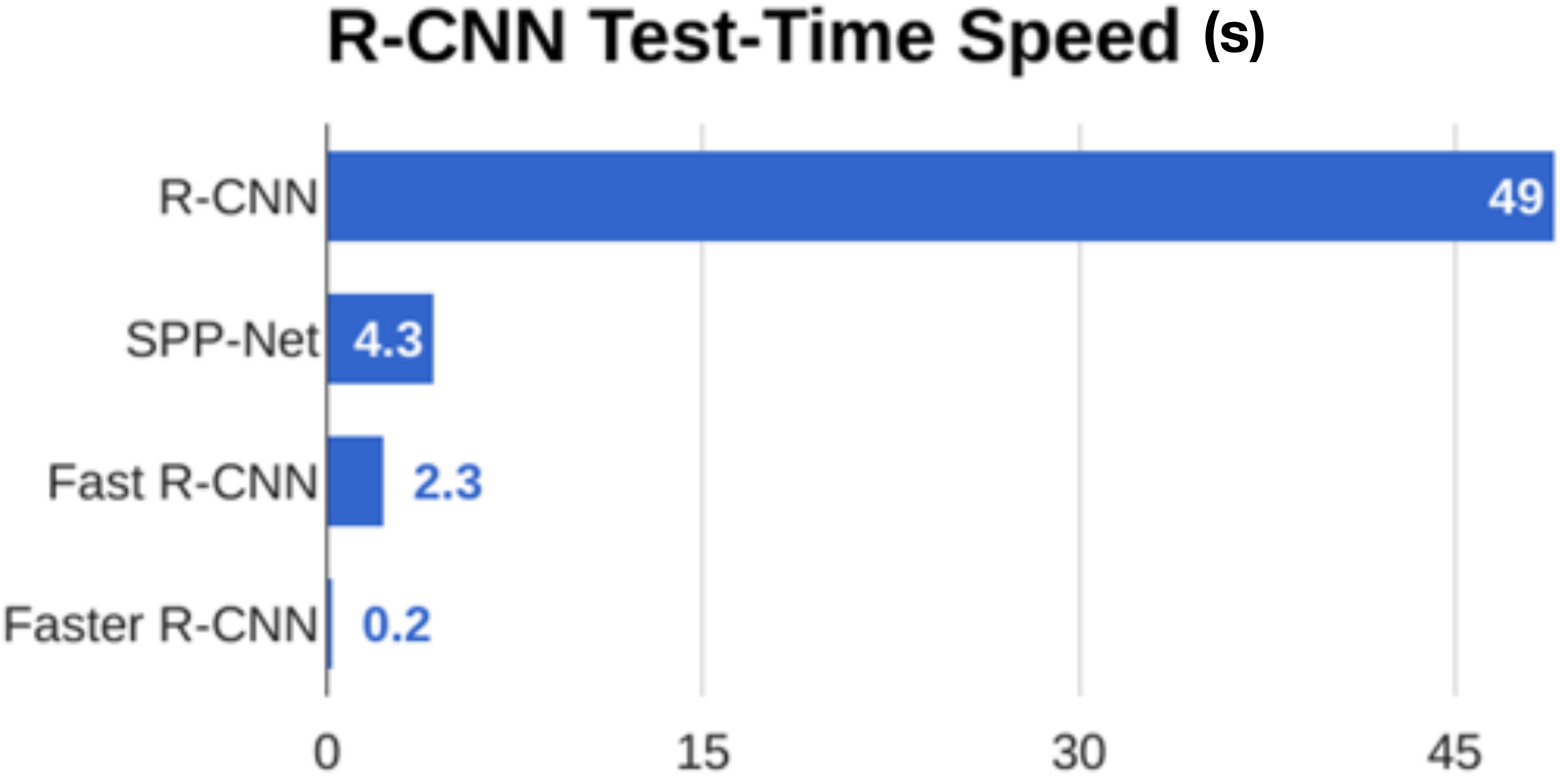
Jointly train four losses:

1. **RPN classification:** anchor box is object / not an object
2. **RPN regression:** predict transform from anchor box to proposal box
3. **Object classification:** classify proposals as background / object class
4. **Object regression:** predict transform from proposal box to object box



Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

Faster R-CNN: Learnable Region Proposals



Extend Faster R-CNN
to Image Segmentation: Mask R-CNN

Classification

"Chocolate Pretzels"

No spatial extent

Semantic Segmentation

Chocolate Pretzels,
Shelf

No objects, just pixels

Object Detection

Instance Segmentation

Flipz, Hershey's, Keese's

Multiple objects

Extend Faster R-CNN
to Instance Segmentation: Mask R-CNN

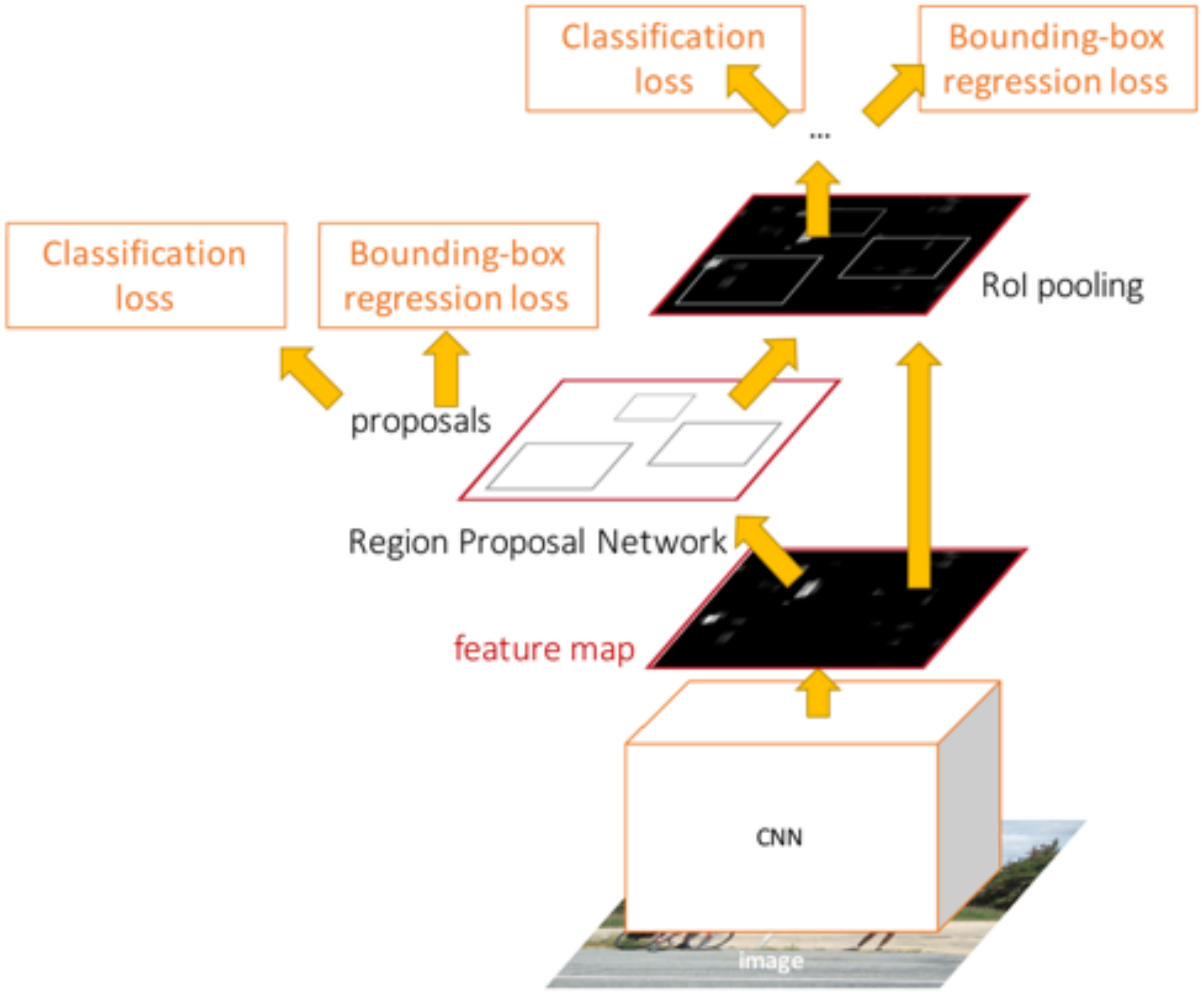
Instance Segmentation
Detect all objects in the image and identify the pixels that belong to each object (Only things!)

Approach
Perform object detection then predict a segmentation mask for each object detected!

Extend Faster R-CNN into Mask R-CNN

Faster R-CNN

1. **Feature Extraction** at the image-level
2. **Regions of Interest** proposal from feature map
3. **In Parallel**
    1. **Object classification:** classify proposals
    2. **Object regression:** predict transforn from proposal box to object box

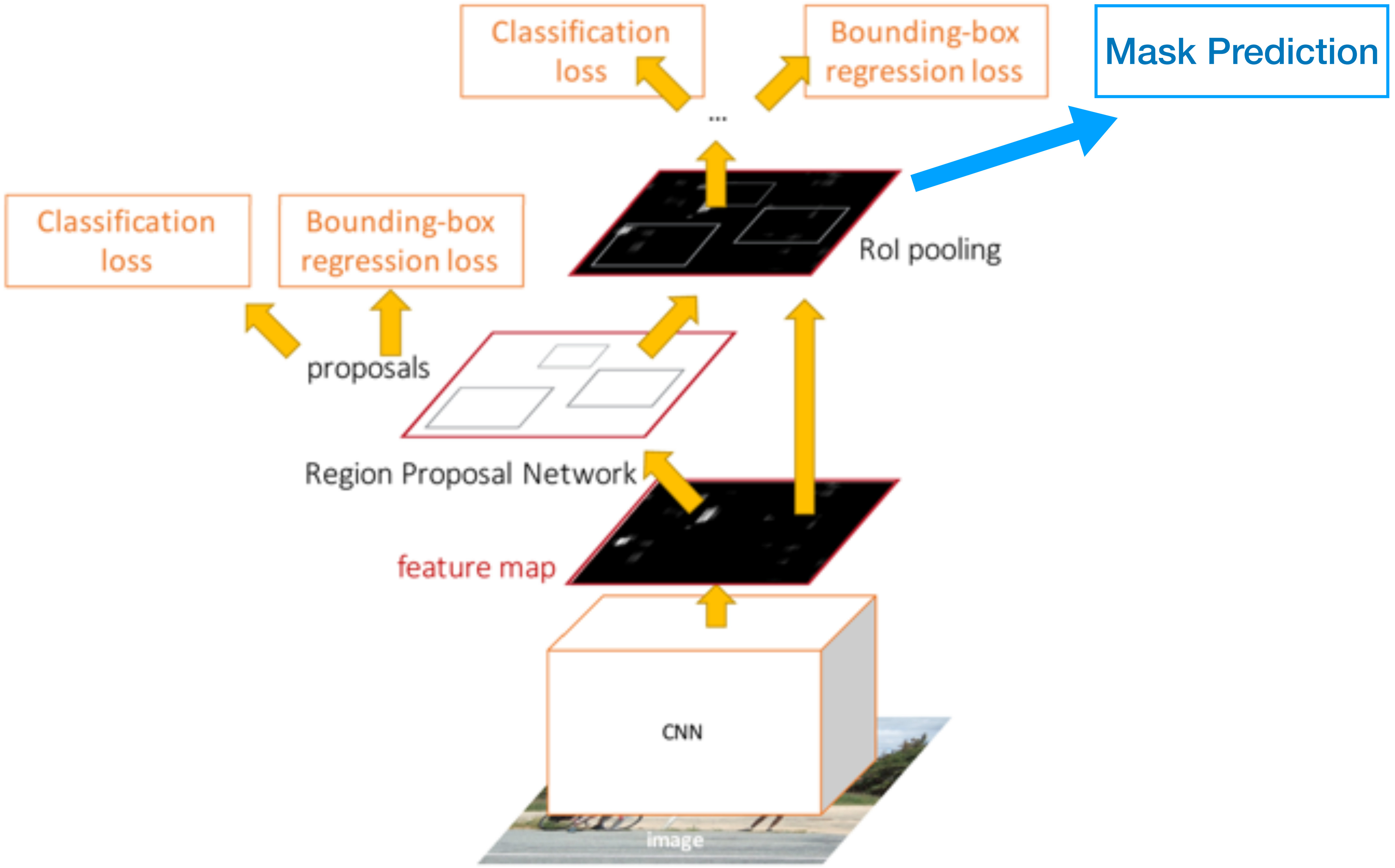


Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick.

Extend Faster R-CNN into Mask R-CNN

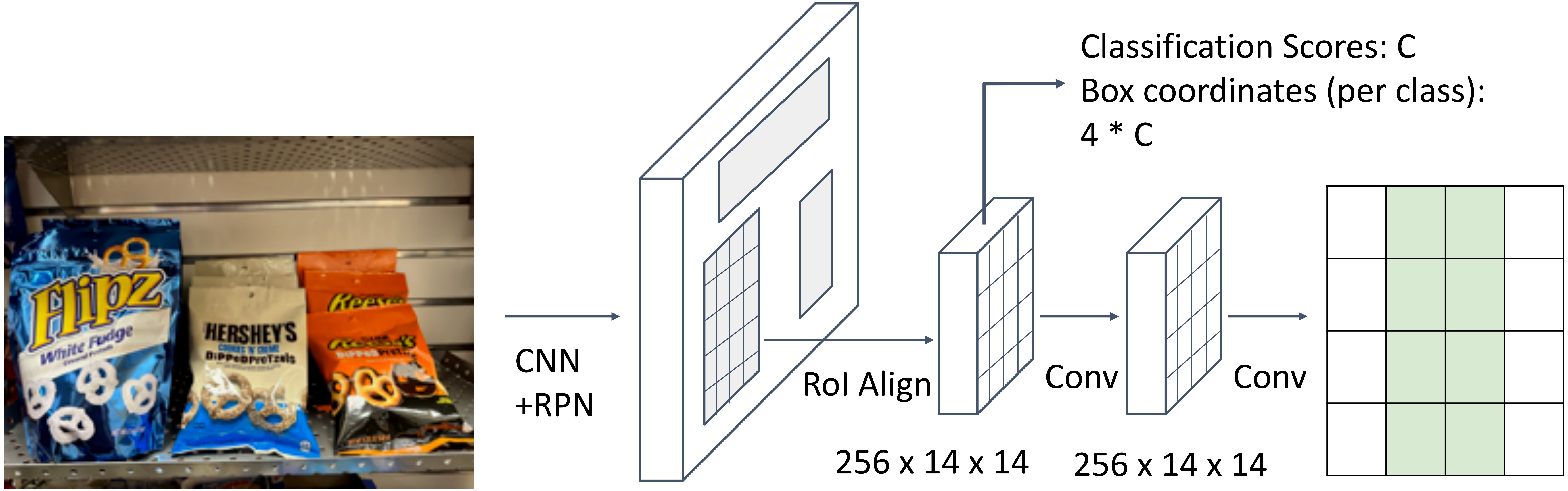
Mask R-CNN

1. **Feature Extraction** at the image-level
2. **Regions of Interest** proposal from feature map
3. **In Parallel**
   a. **Object Classification:** classify proposals
   b. **Object Regression:** predict transform from proposal box to object box
   c. **Mask Prediction:** predict a binary mask for every region



He et al., "Mask R-CNN", ICCV 2017

Mask R-CNN



CNN
+RPN

RoI Align

256 x 14 x 14

Conv

256 x 14 x 14

Conv

Classification Scores: C
Box coordinates (per class):
4 * C

Predict a mask for
each of C classes:
C x 28 x 28

He et al., “Mask R-CNN”, ICCV 2017

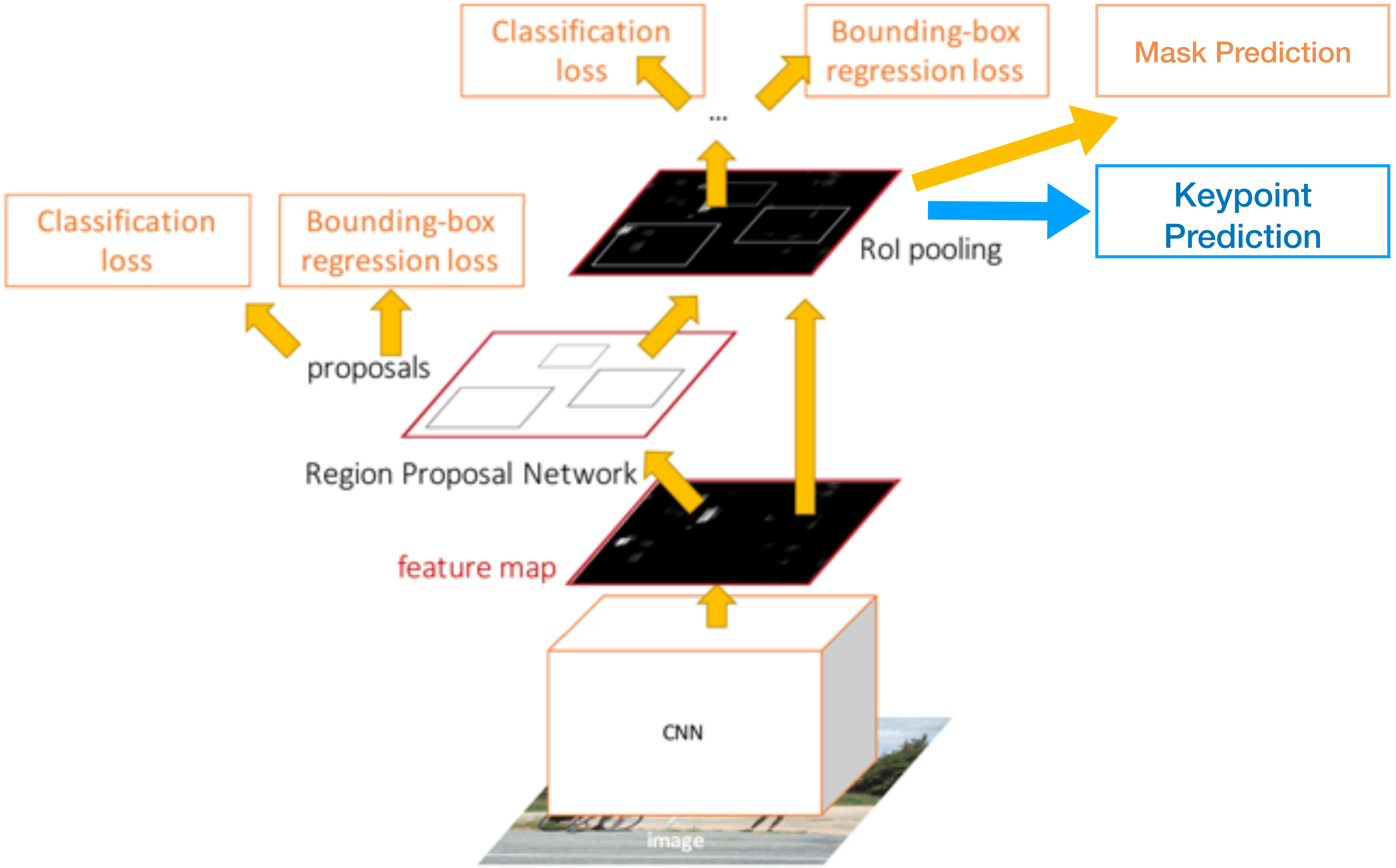
Mask R-CNN: Very Good Results!

He et al., “Mask R-CNN”, ICCV 2017

Mask R-CNN for Human Pose Estimation

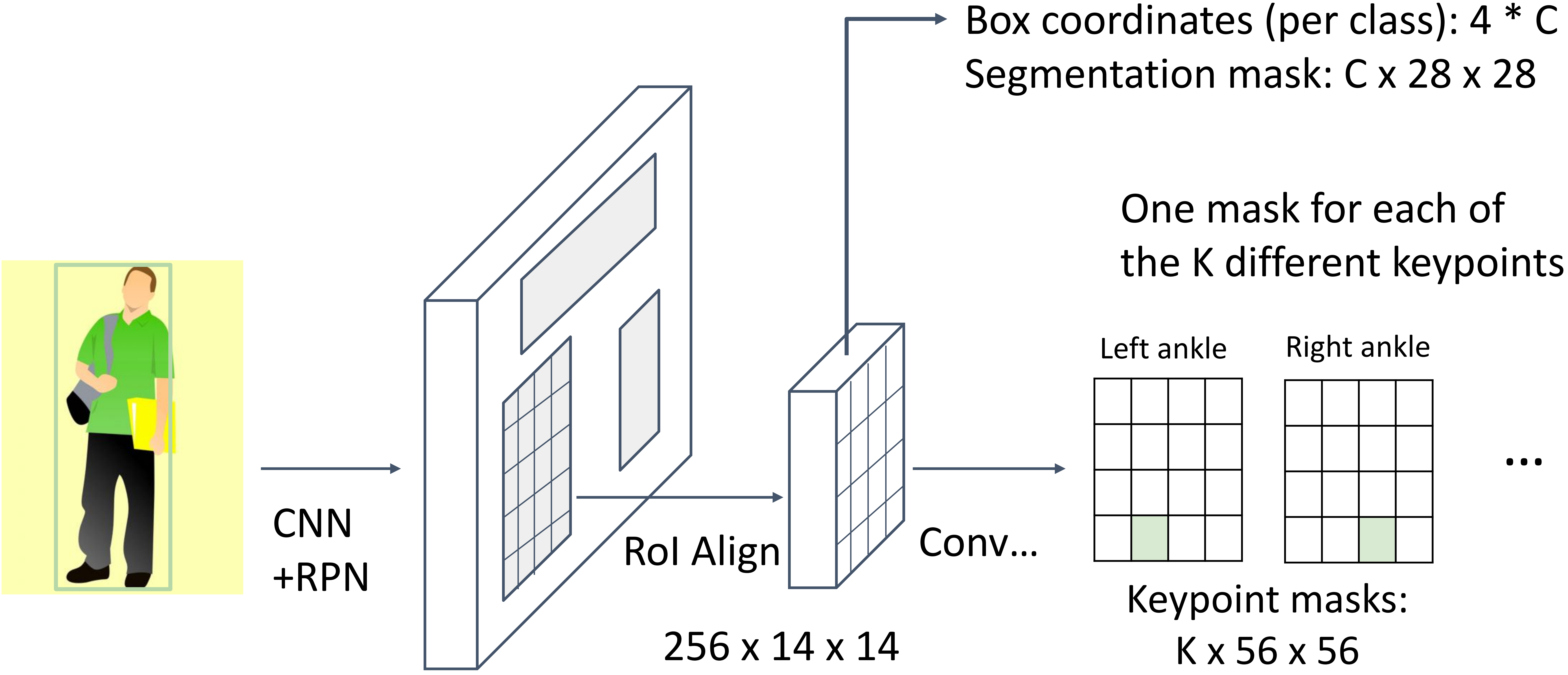
Mask R-CNN

1. **Feature Extraction** at the image-level
2. **Regions of Interest** proposal from feature map
3. **In Parallel**
    a. **Object Classification:** classify proposals
    b. **Object Regression:** predict transform from proposal box to object box
    c. **Mask Prediction:** predict a binary mask for every region
    d. **Keypoint Prediction:** predict binary mask for human key points



He et al., "Mask R-CNN", ICCV 2017

Mask R-CNN for Human Pose Estimation



Classification Scores: C
Box coordinates (per class): 4 * C
Segmentation mask: C x 28 x 28

CNN
+RPN

RoI Align

256 x 14 x 14

Conv…

One mask for each of
the K different keypoints

Left ankle

Right ankle

…

Keypoint masks:
K x 56 x 56

Ground-truth has one “pixel” turned on per keypoint. Train with softmax loss

He et al., “Mask R-CNN”, ICCV 2017

Mask R-CNN for Human Pose Estimation

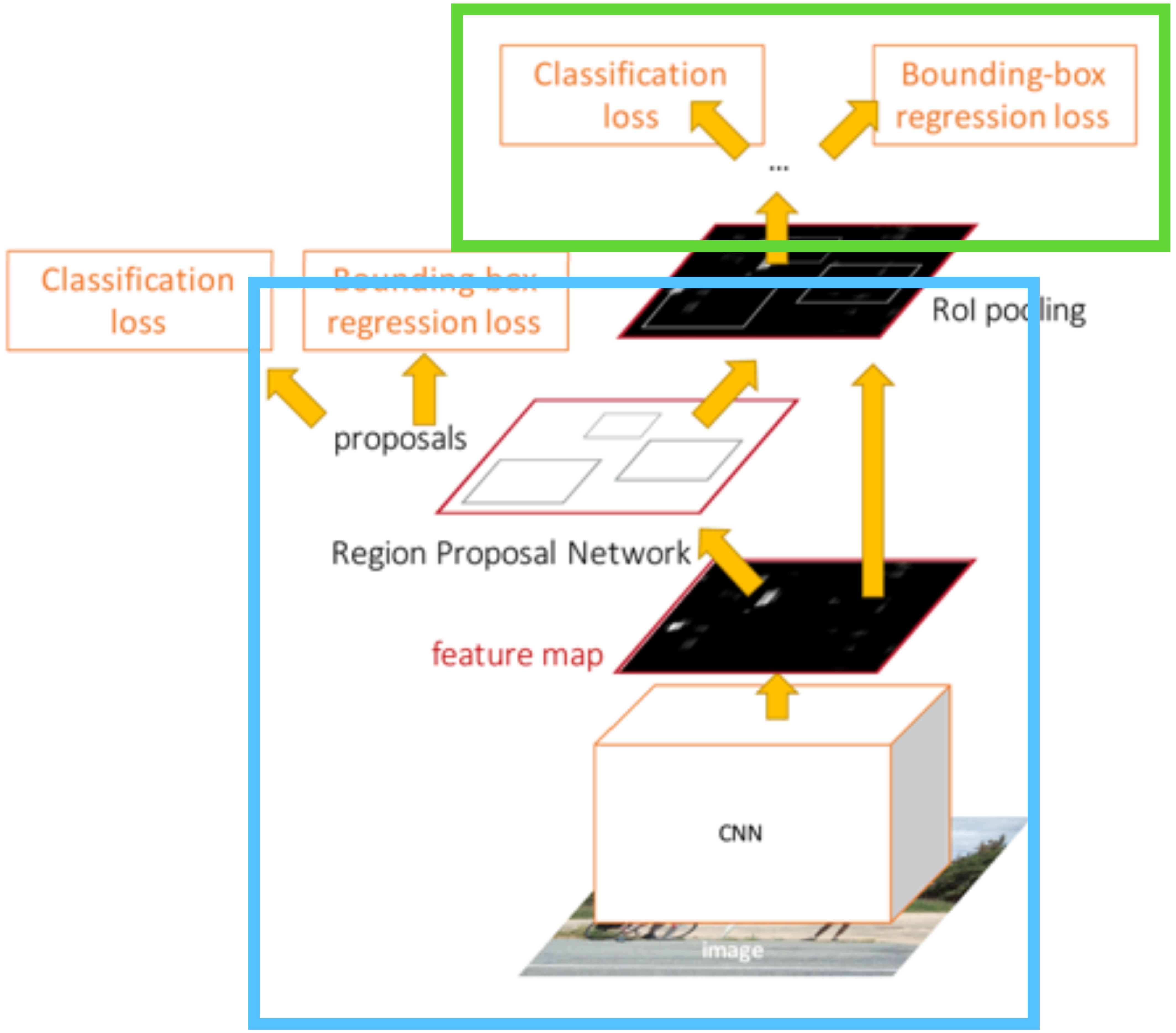
He et al., “Mask R-CNN”, ICCV 2017

Two Stage vs One Stage Detectors

Faster R-CNN is a two-stage object detector

First stage: Run once per image
- Backbone Network
- Region Proposal Network

Second stage: Run once per region
- Crop features: RoI pool / align
- Predict Object Class
- Prediction bbox offset



Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015
Figure copyright 2015, Ross Girshick.

Next Time:

RGB-D Perception and Network Architectures

DeepRob

Lecture 13
Object Detectors and Segmentation

University of Michigan and University of Minnesota

Slides adapted from Justin Johnson's Deep Learning for Computer Vision course