Object Detection

Longbin Jin

Artificial Inteligence
& Computer Vision
Laboratory

Slide reference: Stanford University cs231n, Fei-Fei Li

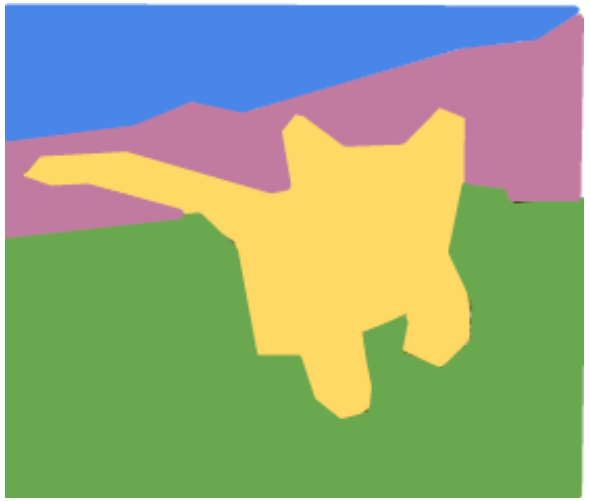
The Generalized R-CNN Framework for Object Detection, Ross Girshick, ICCV 2019 tutorial

Computer Vision Tasks

Classification

CAT

Semantic Segmentation



GRASS, CAT
TREE, SKY

Object Detection

DOG, DOG, CAT

Instance Segmentation

DOG, DOG, CAT

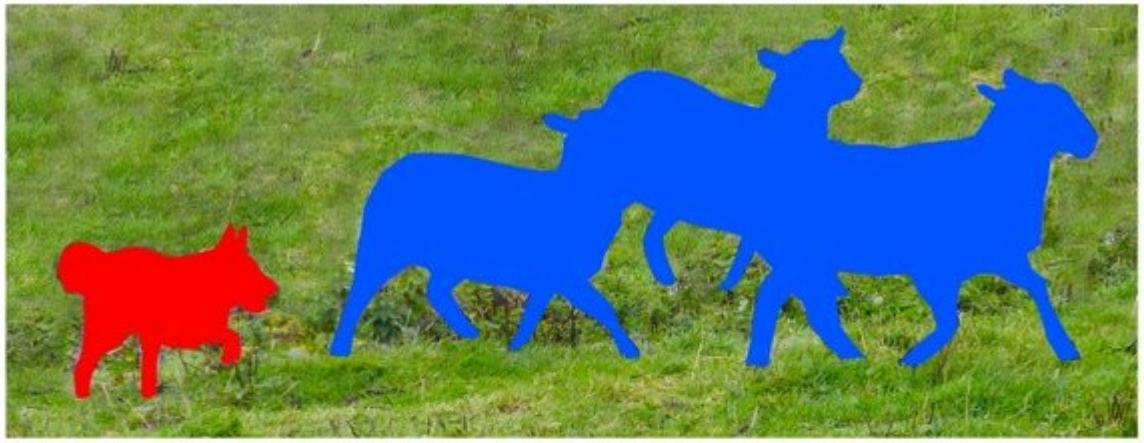
No spatial extent

No objects, just pixels

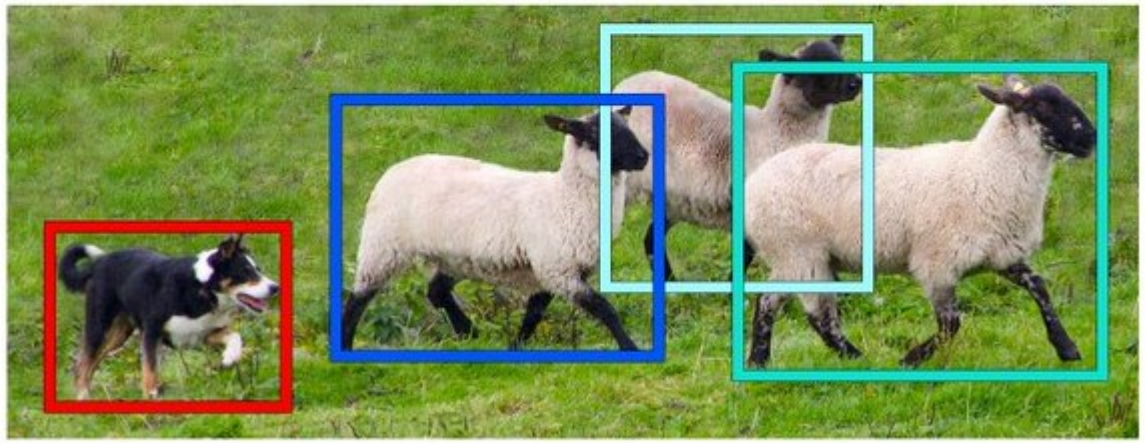
Multiple Object

Computer Vision Tasks

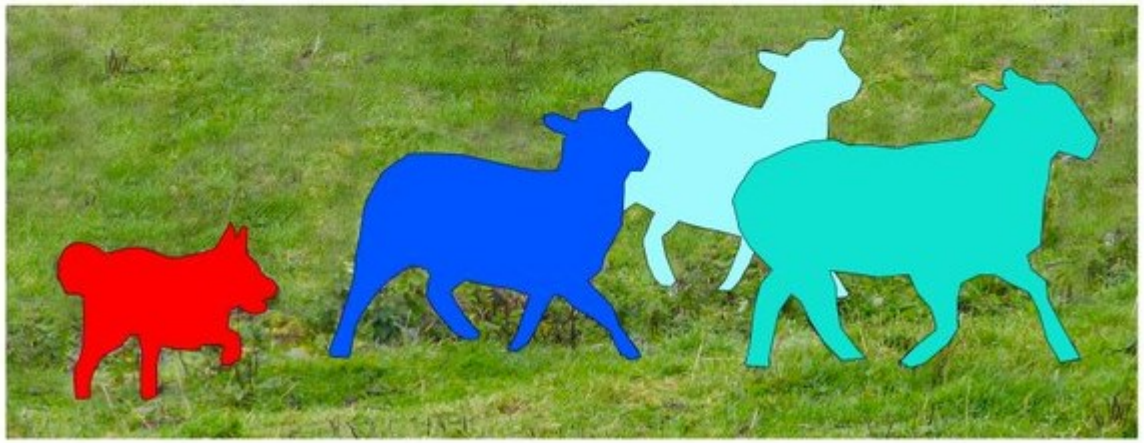
Image Recognition



Semantic Segmentation

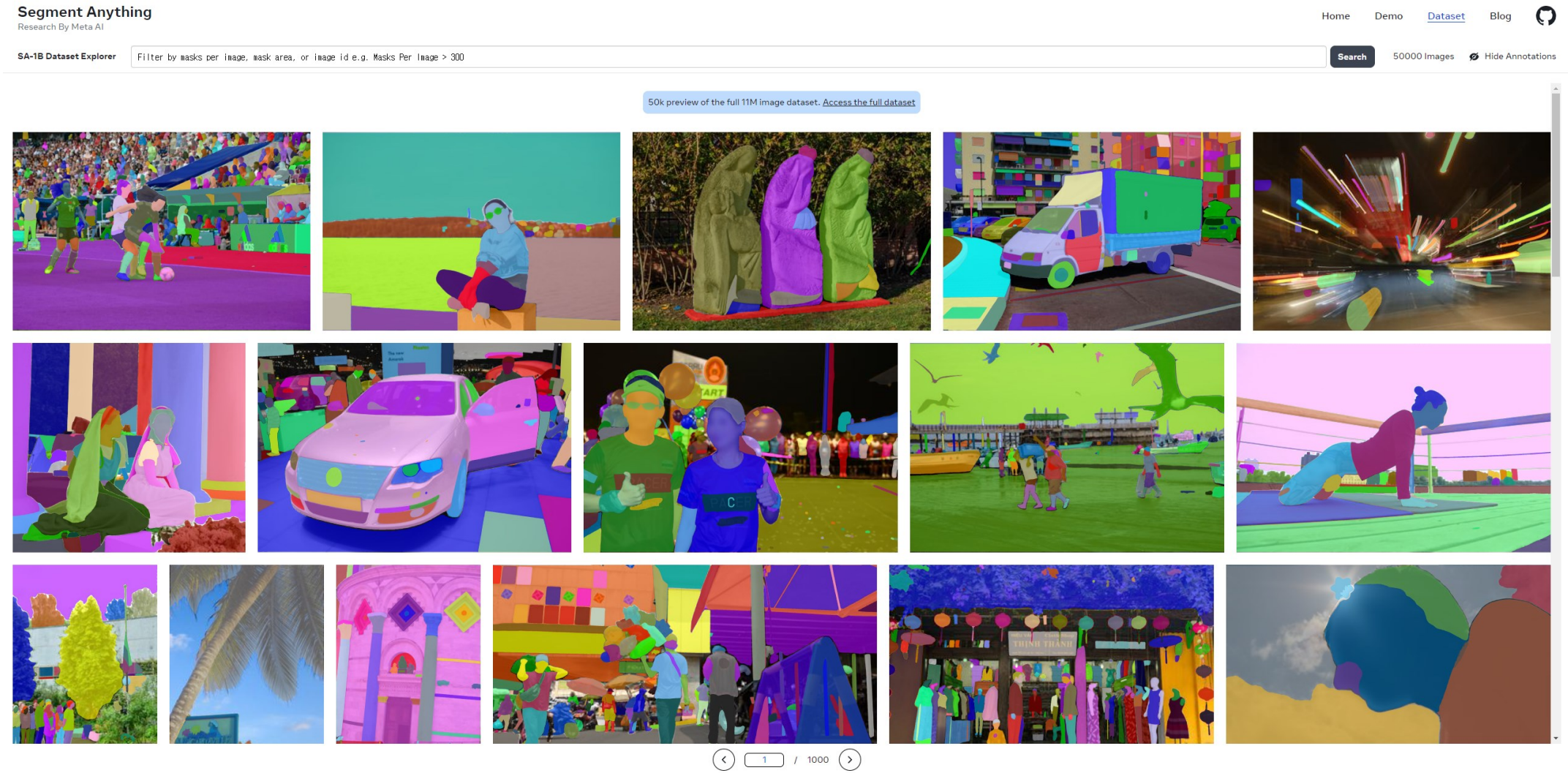


Object Detection



Instance Segmentation

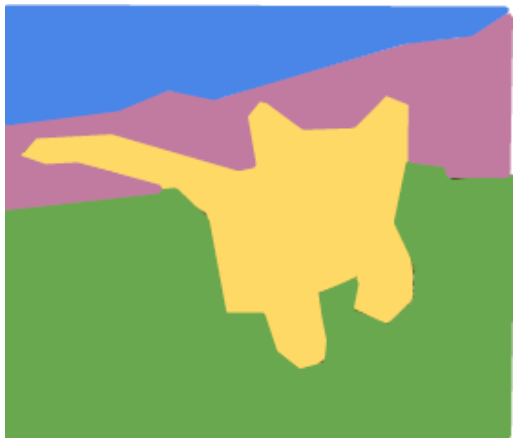
Segment Anything Model (SAM) 2023



https://segment-anything.com/

Semantic Segmentation

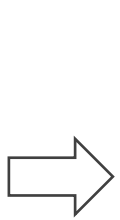
Train

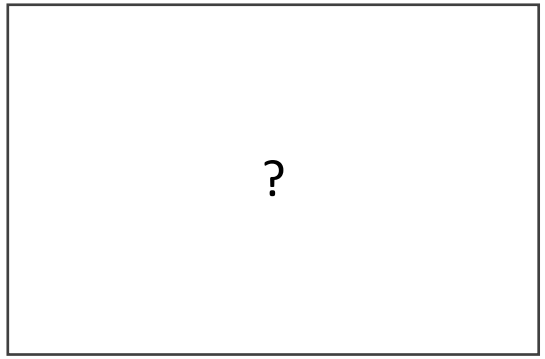


GRASS, CAT
TREE, SKY

Paired training data: for each training image, each pixel is labeled with a semantic category

Test





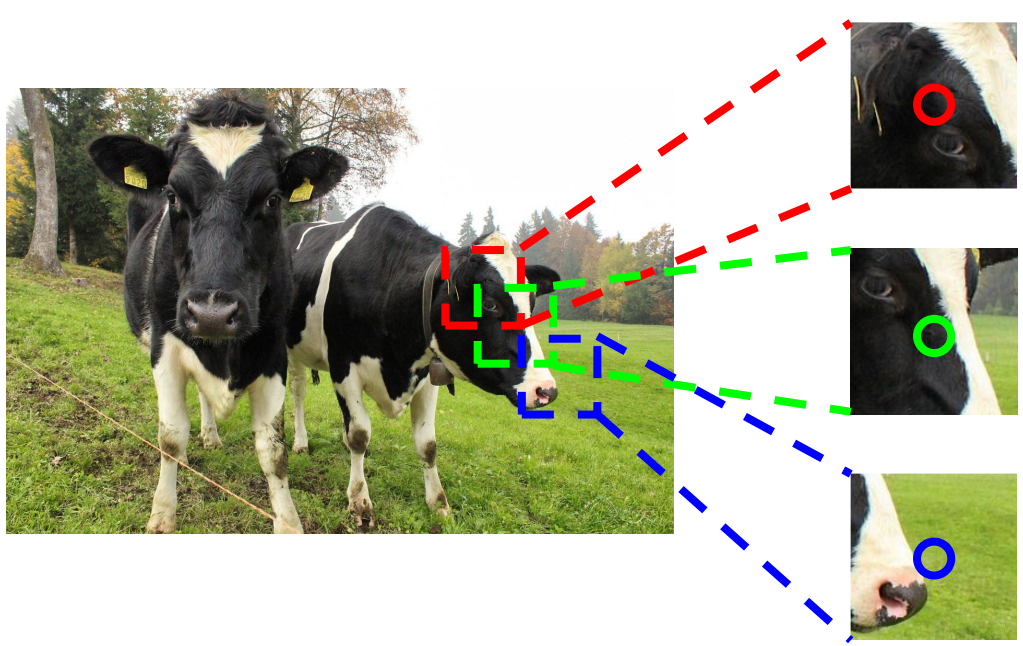
At test time, classify each pixel of a new image.

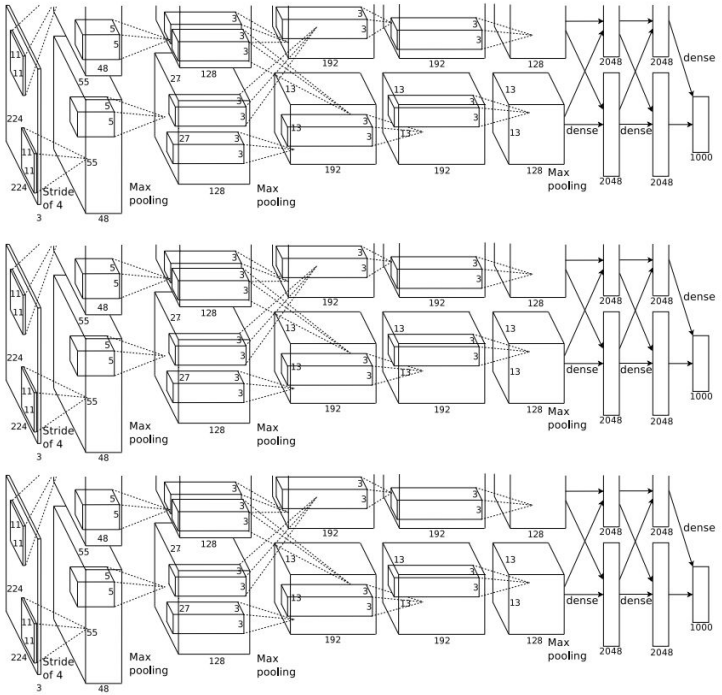
Semantic Segmentation Idea: Sliding Window

Impossible to classify without context

Q: how do we include context?

Semantic Segmentation Idea: Sliding Window





Cow

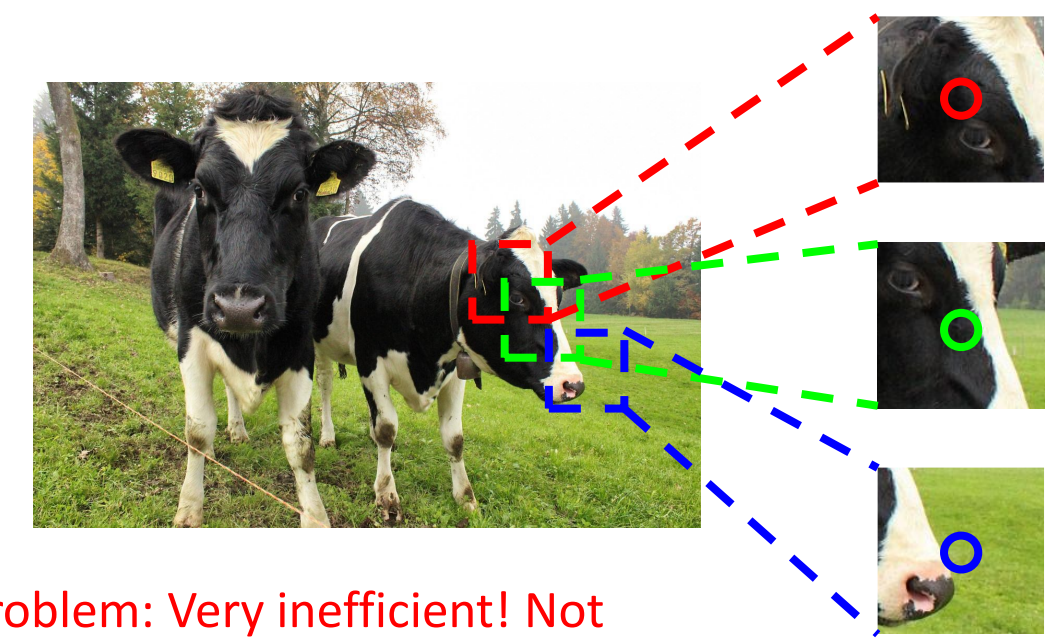
Cow

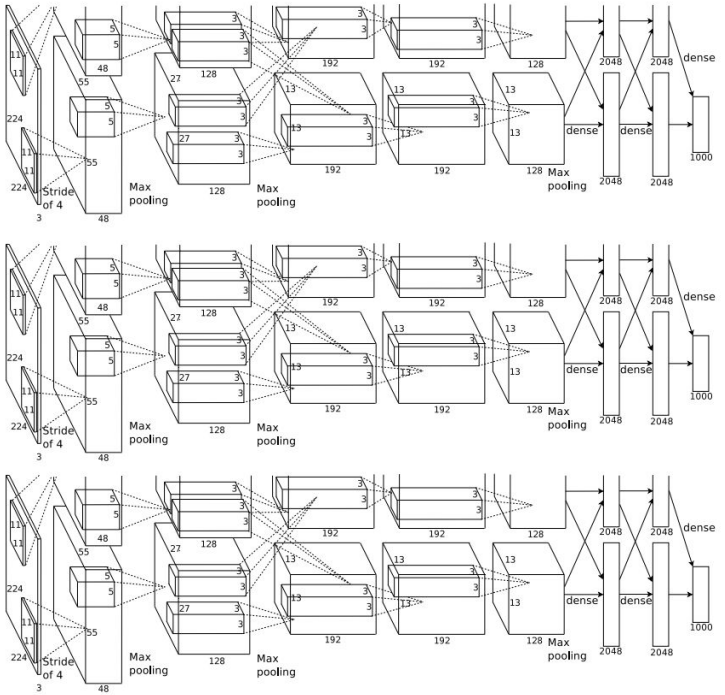
Grass

Q: how do we model this?

Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013
Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014

Semantic Segmentation Idea: Sliding Window





Cow

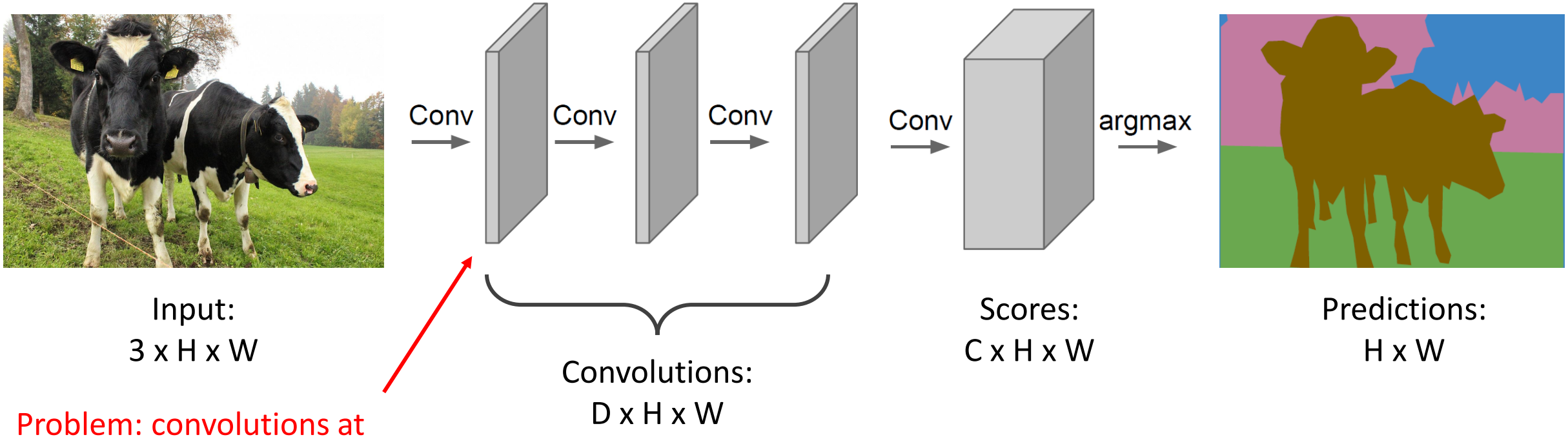
Cow

Grass

Problem: Very inefficient! Not reusing shared features between overlapping patches

Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013
Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014

Semantic Segmentation Idea: Fully Convolutional



Input:
3 x H x W

Convolutions:
D x H x W

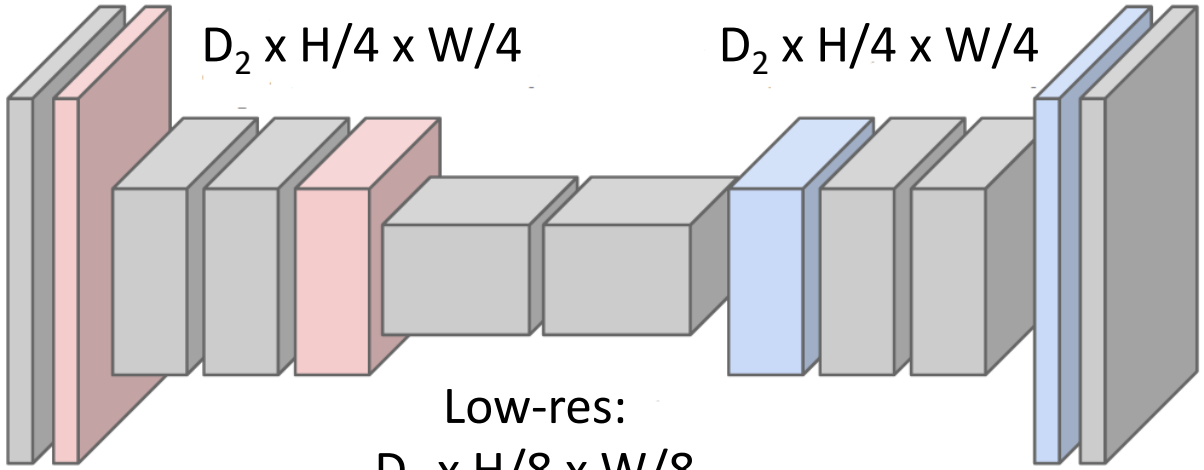
Scores:
C x H x W

Predictions:
H x W

Problem: convolutions at original image resolution will be very expensive ...

Semantic Segmentation Idea: Fully Convolutional

Input:
3 x H x W



Predictions:
H x W

Downsampling:
Pooling, strided convolution

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Upsampling:
??

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015

In-Network upsampling: “Unpooling”

Nearest Neighbor

1	2
3	4

Input: 2 x 2

→

1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

Output: 4 x 4

“Bed of Nails”

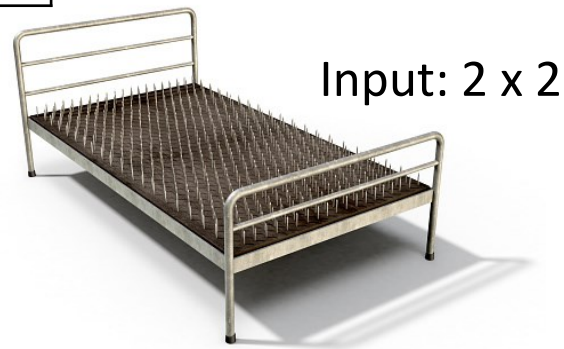
1	2
3	4

Input: 2 x 2

→

1	0	2	0
0	0	0	0
3	0	4	0
0	0	0	0

Output: 4 x 4



Semantic Segmentation Idea: Fully Convolutional

Max Pooling
Remember which element was max!

1	2	6	3
3	5	2	1
1	2	2	1
7	3	4	8

→

5	6
7	8

→ Rest of the network … →

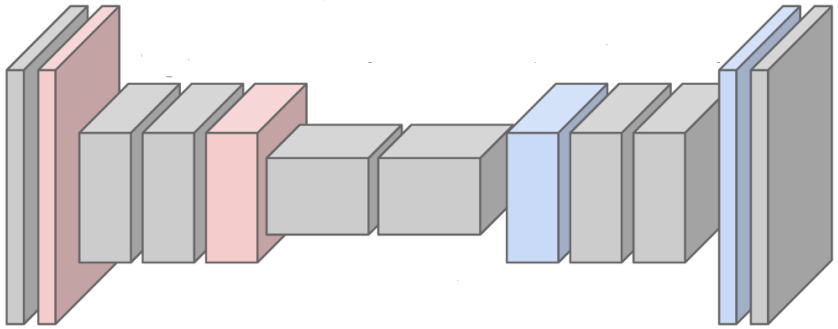
1	2
3	4

→

Max Unpooling
Use positions from pooling layer

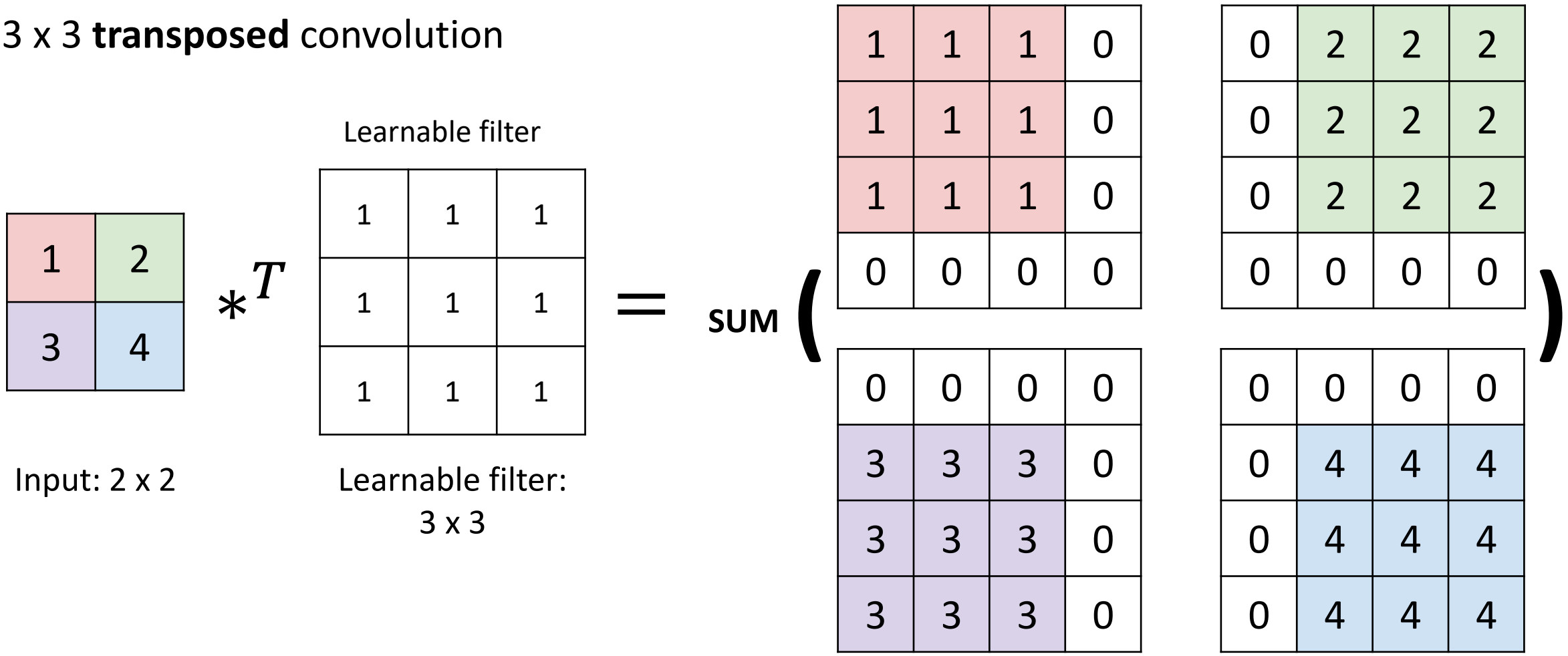
0	0	2	0
0	1	0	0
0	0	0	0
3	0	0	4

Corresponding pairs of downsampling and upsampling layers



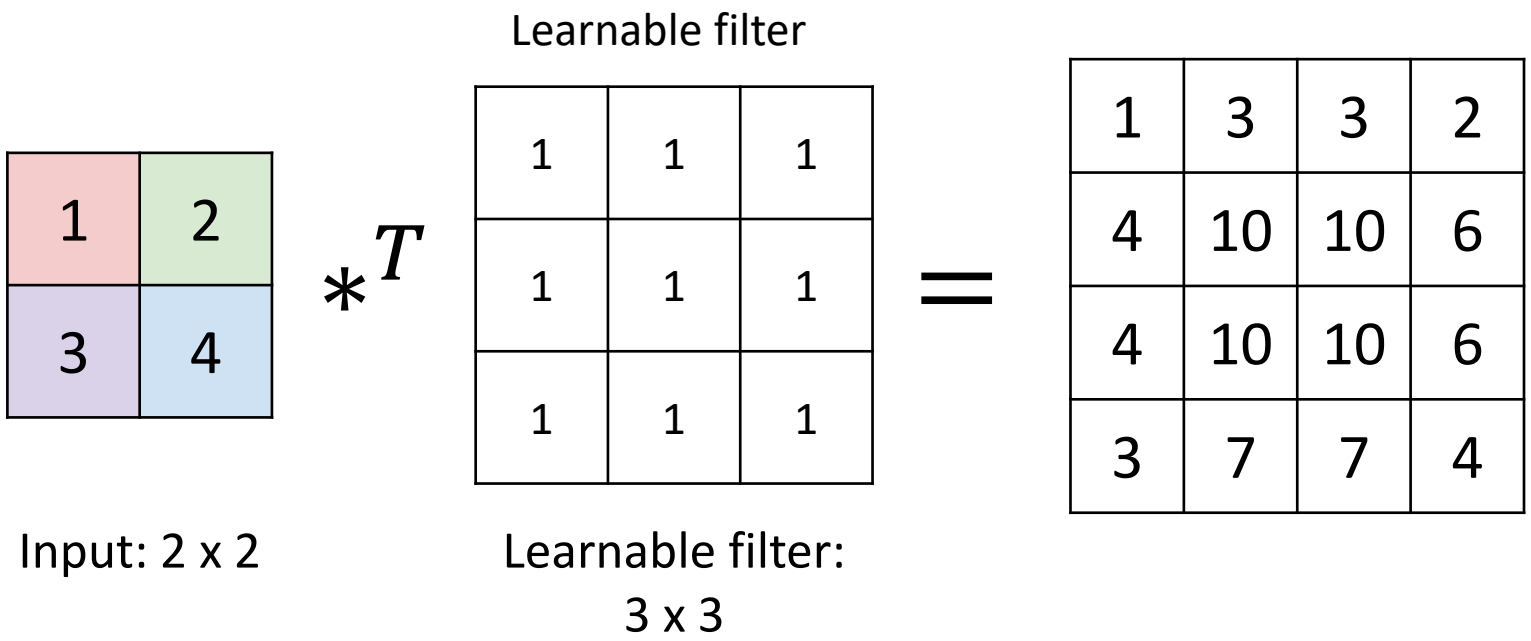
Learnable Upsampling

3 x 3 **transposed** convolution



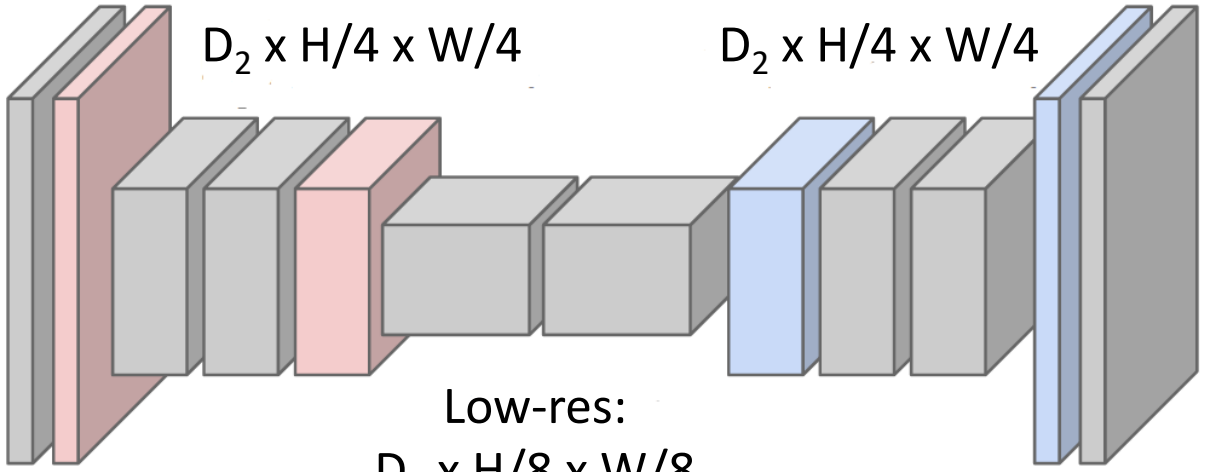
Learnable Upsampling

3 x 3 **transposed** convolution



Semantic Segmentation Idea: Fully Convolutional

Input:
3 x H x W



Predictions:
H x W

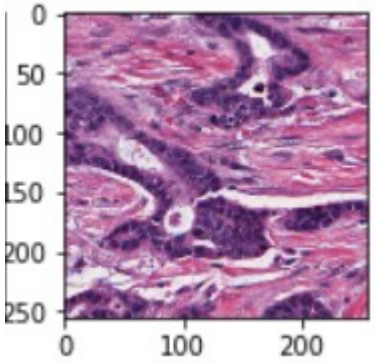
Downsampling:
Pooling, strided convolution

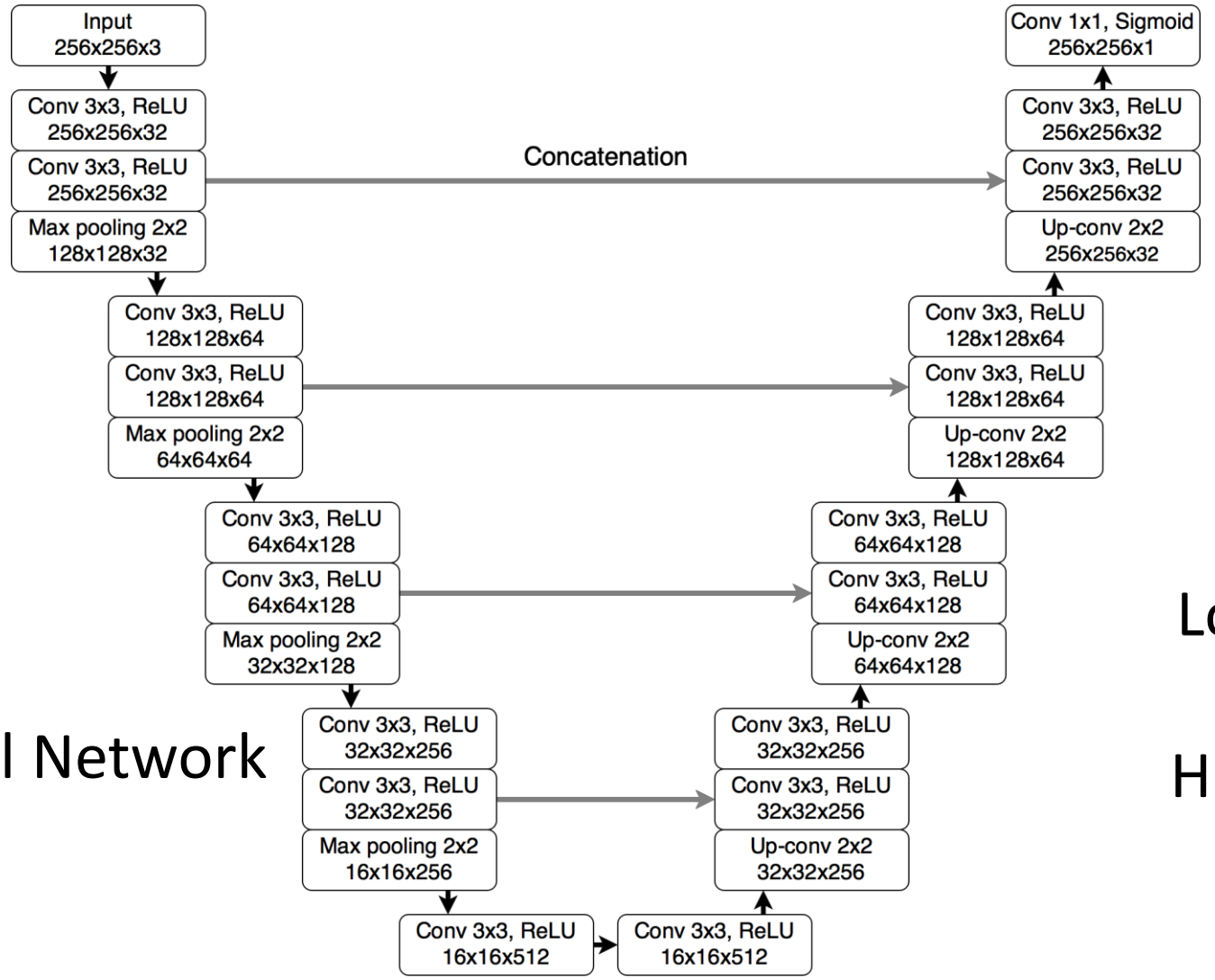
Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

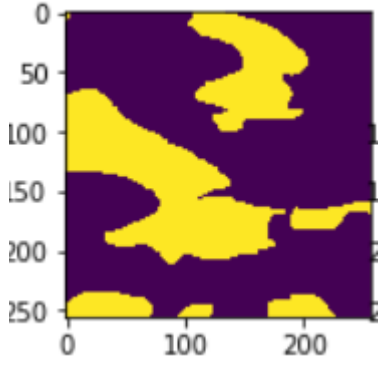
Upsampling:
Unpooling or strided transposed convolution

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015

Unet







Fully Convolutional Network
FCN

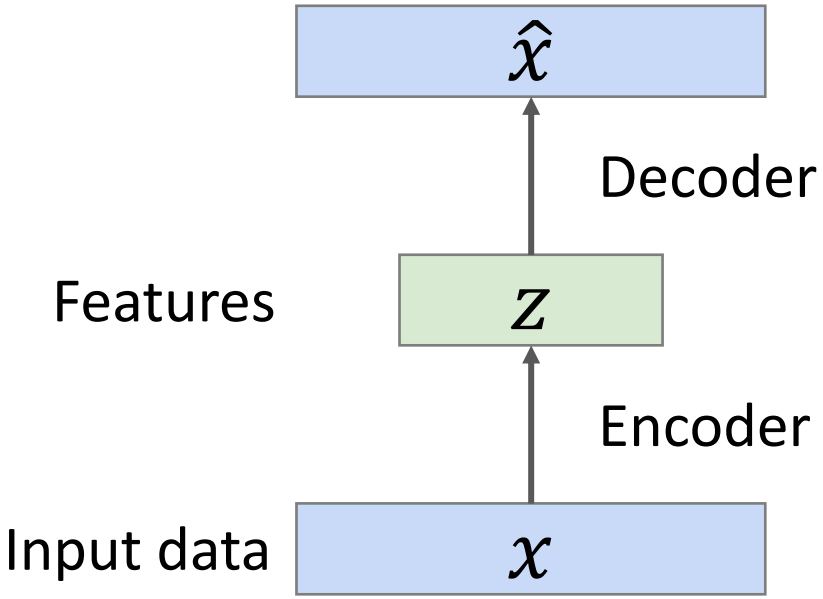
Low-level feature
+
High-level feature

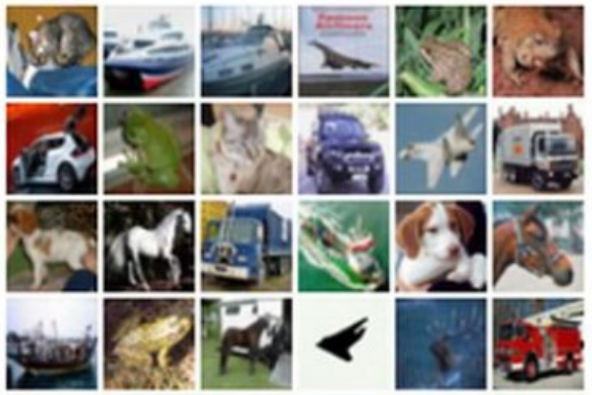
Olaf et al., U-Net: Convolutional Networks for Biomedical Image Segmentation, 2015

AutoEncoder (AE)

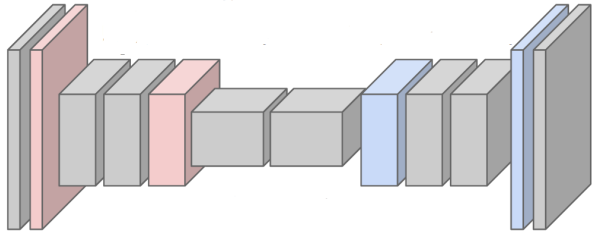
- Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data

Want features to capture meaningful factors of variation in data automatically





AutoEncoder (AE)



How to learn this feature representation?

Reconstructed input data

$\hat{x}$

Decoder

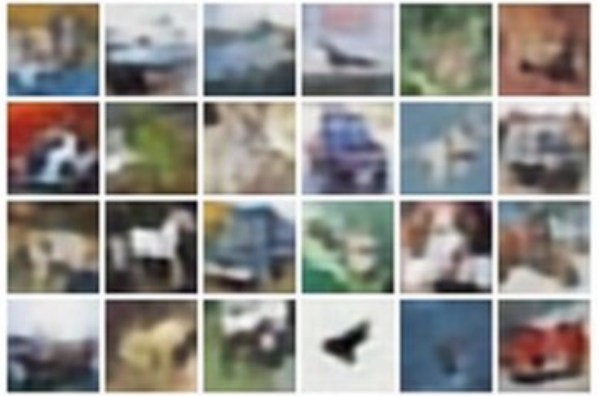
Features z

Encoder

Input data x

Train such that features can be used to reconstruct original data “Autoencoding” encoding input itself

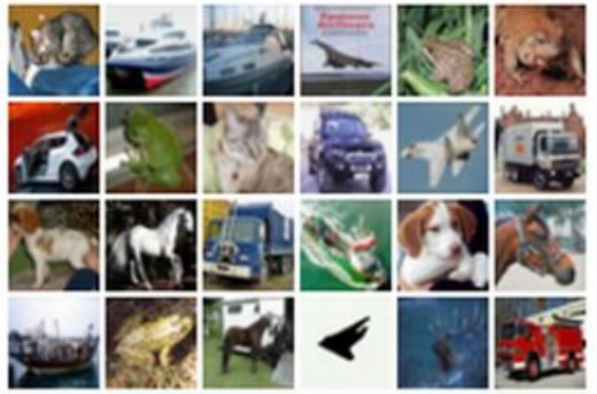
Reconstructed data



Decoder: 4-layer upconv

Encoder: 4-layer conv

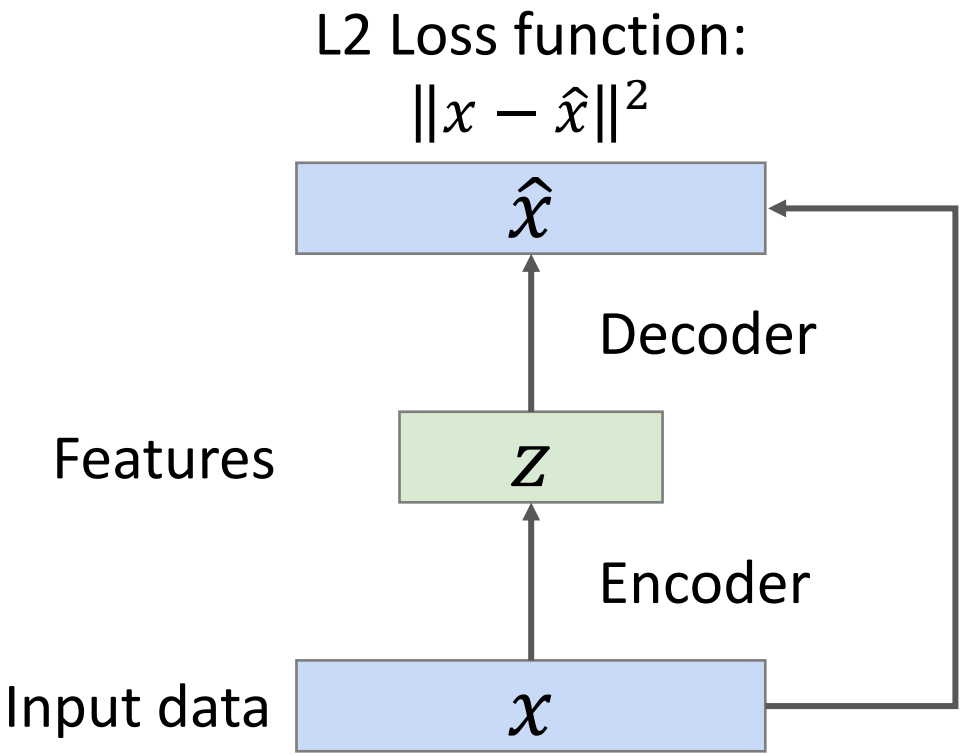
Input data



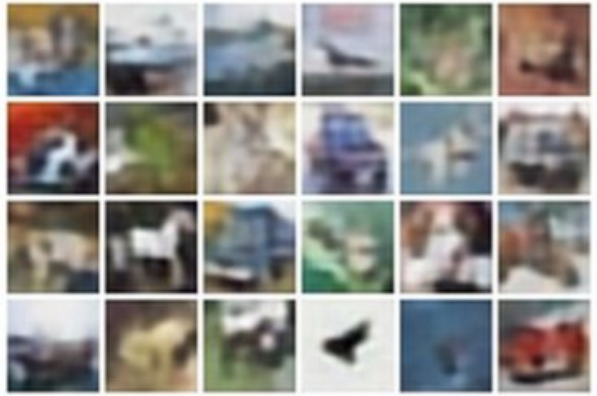
AutoEncoder (AE)

Doesn't use labels!

Train such that features can be used to reconstruct original data

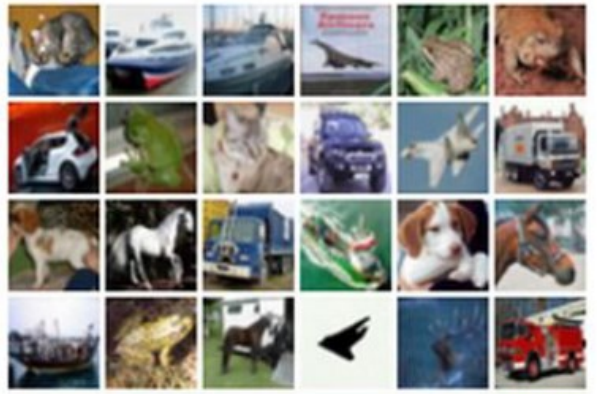


Reconstructed data



Decoder: 4-layer upconv
Encoder: 4-layer conv

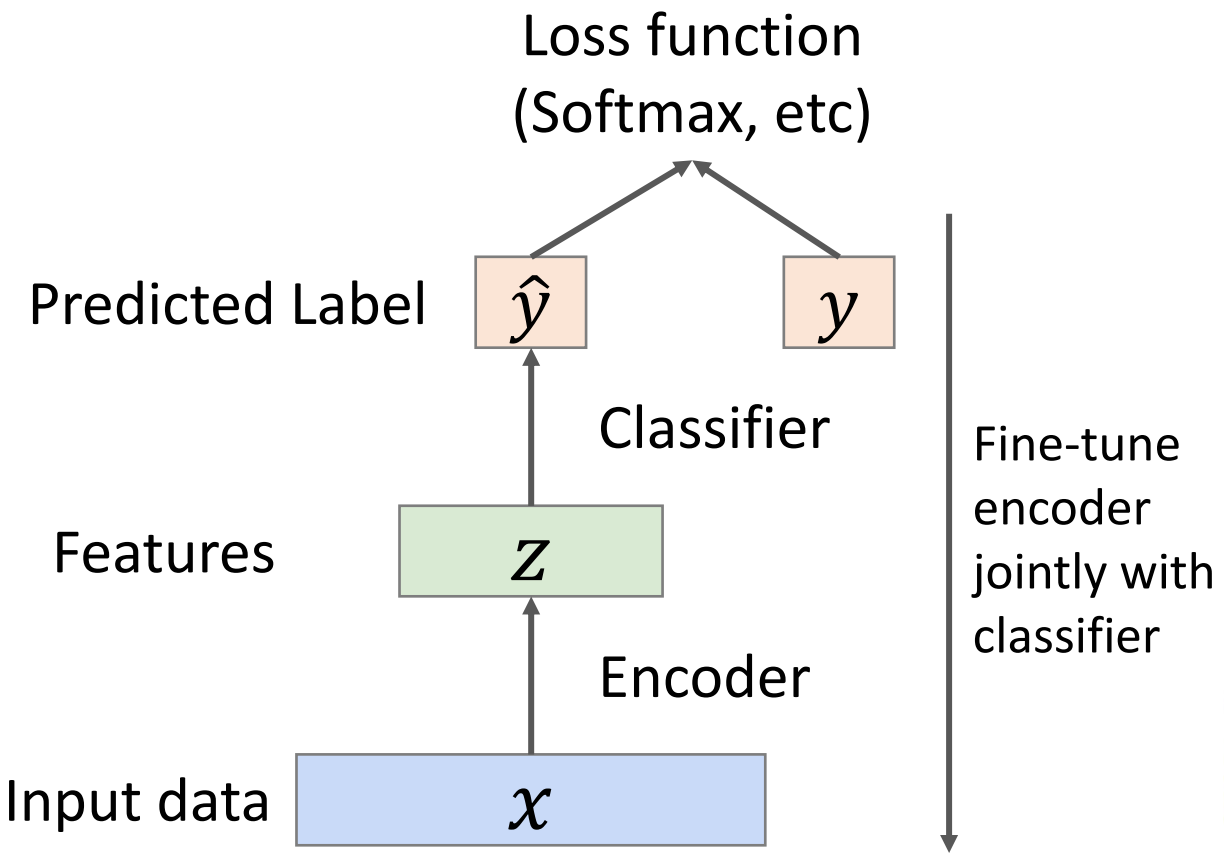
Input data



AutoEncoder (AE)

Transfer from large, unlabeled dataset to small, labeled dataset.

Encoder can be used to initialize a supervised model



Fine-tune encoder jointly with classifier

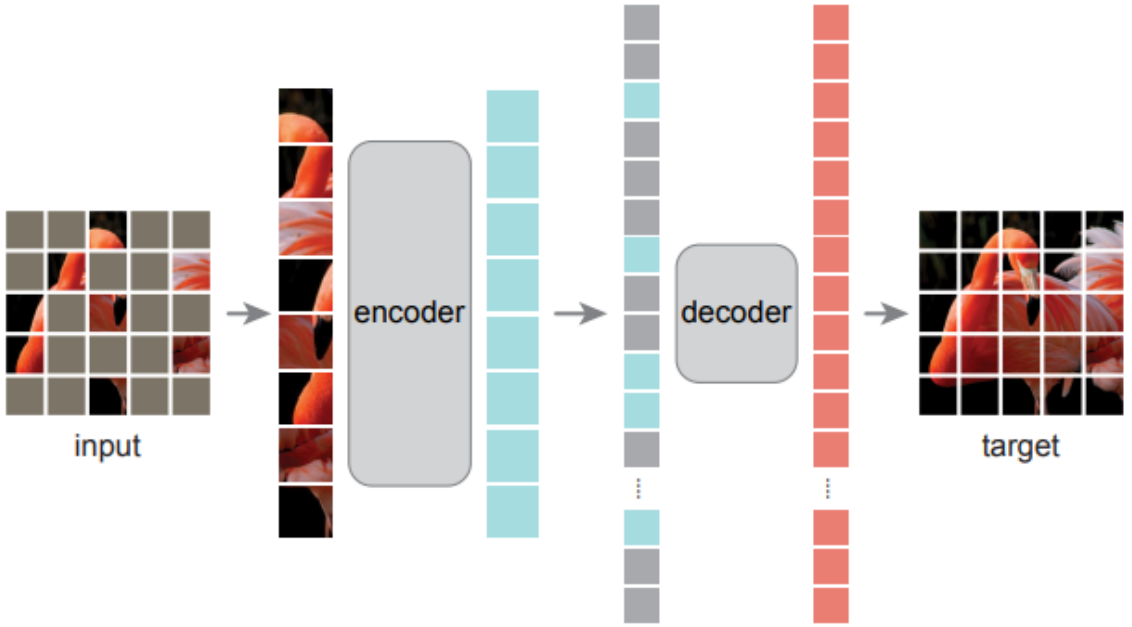
bird plane
dog deer truck

Train for final task (sometimes with small data)

Masked Autoencoder (MAE)

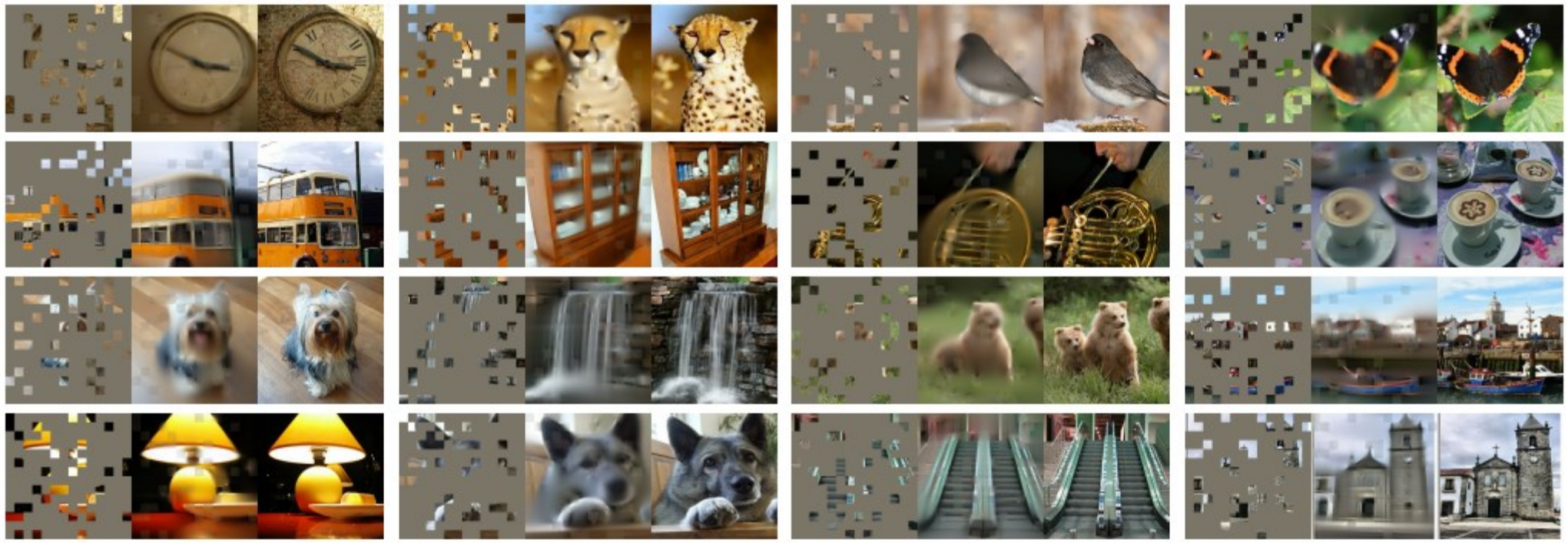
- Pretext tasks
  - Mask random patches of the input image and reconstruct the missing pixels.

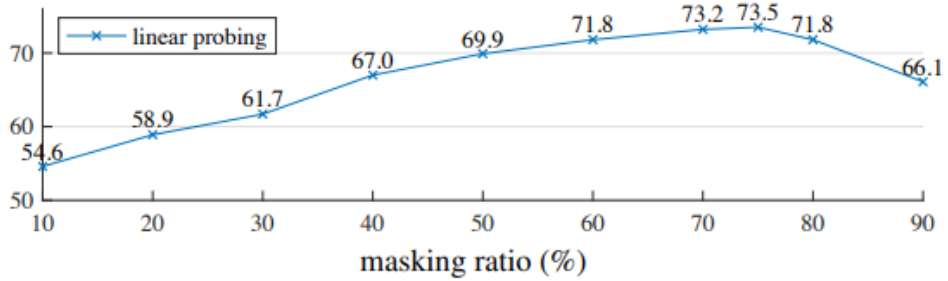
MAE encoder & decoder:
Vision Transformer



He et al. Masked Autoencoders Are Scalable Vision Learners 2021

Masked Autoencoder (MAE)



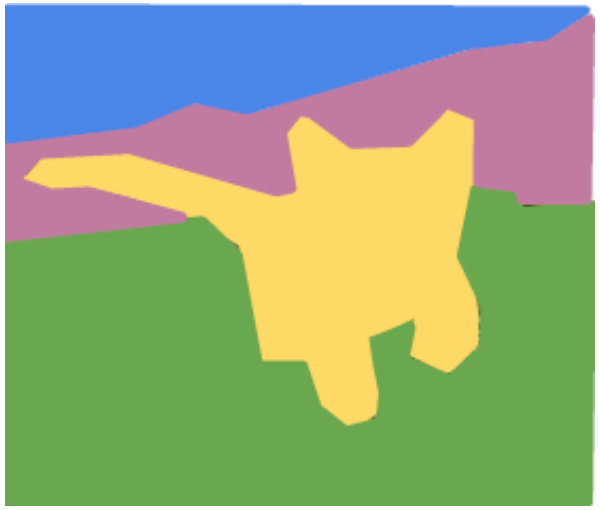


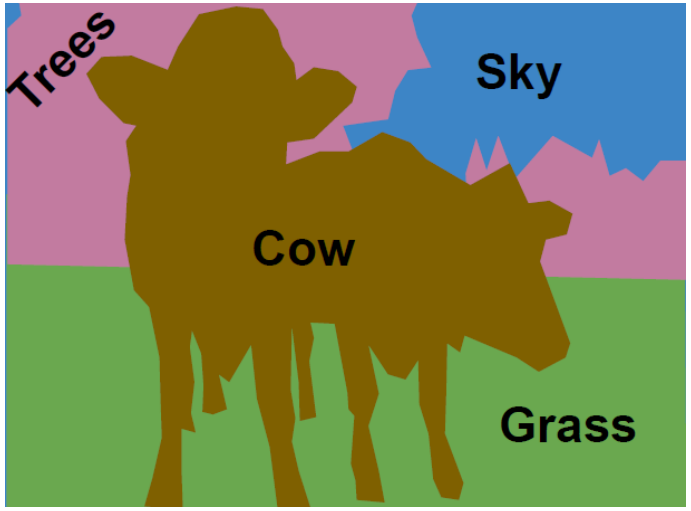
He et al. Masked Autoencoders Are Scalable Vision Learners 2021

Semantic Segmentation

Label each pixel in the image with a category label

Don’t differentiate instances, only care about pixels



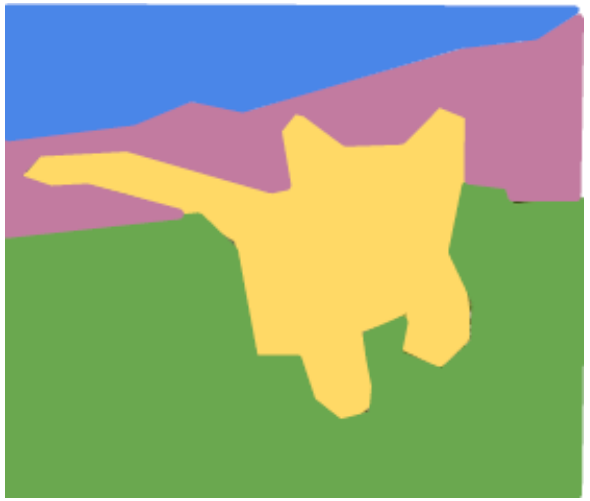


Computer Vision Tasks

Classification

CAT

Semantic Segmentation



GRASS, CAT
TREE, SKY

Object Detection

DOG, DOG, CAT

Instance Segmentation

DOG, DOG, CAT

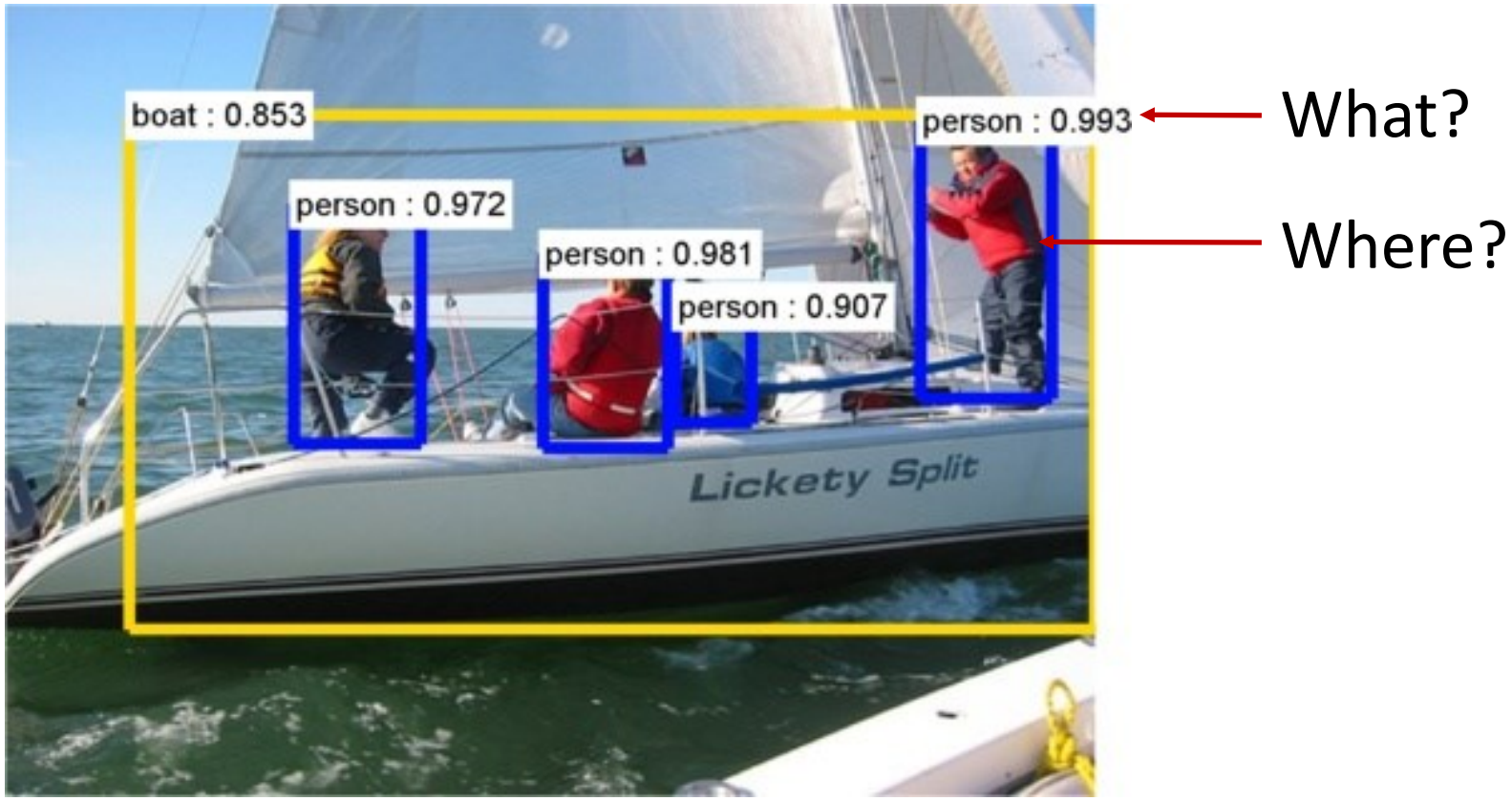
No spatial extent

No objects, just pixels

Multiple Object

Object Detection

What objects are in an image and where are they?



Steady Progress on Boxes and Masks

- R-CNN [Girshick et al. 2014]
- SPP-net [He et al. 2014]
- Fast R-CNN [Girshick. 2015]
- Faster R-CNN [Ren et al. 2015]
- R-FCN [Dai et al. 2016]
- Feature Pyramid Networks + Faster R-CNN [Lin et al. 2017]
- Mask R-CNN [He et al. 2017]
- Training with Large Minibatches (MegDet) [Peng, Xiao, Li, et al. 2017]
- Cascade R-CNN [Cai & Vasconcelos 2018]
- …

Steady Progress on Boxes and Masks

2014

Rich feature hierarchies for accurate object detection and semantic segmentation
Tech report (v5)

Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik
UC Berkeley

2015

Fast R-CNN

Ross Girshick
Microsoft Research

2015

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

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

2017

Mask R-CNN

Kaiming He Georgia Gkioxari Piotr Dollár Ross Girshick
Facebook AI Research (FAIR)

2021

Masked Autoencoders Are Scalable Vision Learners

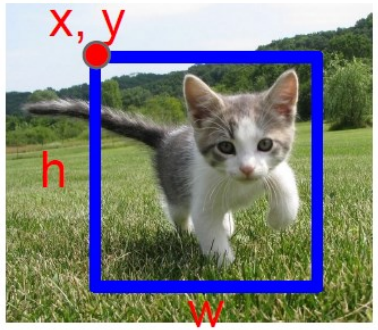
Kaiming He*,† Xinlei Chen* Saining Xie Yanghao Li Piotr Dollár Ross Girshick
*equal technical contribution †project lead
Facebook AI Research (FAIR)

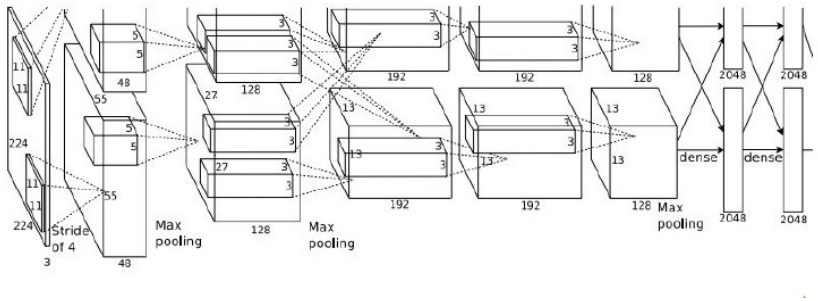
2023

Segment Anything

Alexander Kirillov[1,2,4] Eric Mintun[2] Nikhila Ravi[1,2] Hanzi Mao[2] Chloe Rolland[3] Laura Gustafson[3]
Tete Xiao[3] Spencer Whitehead Alexander C. Berg Wan-Yen Lo Piotr Dollár[4] Ross Girshick[4]
[1]project lead [2]joint first author [3]equal contribution [4]directional lead
Meta AI Research, FAIR

Object Detection: Single Object





Fully Connected: 4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01

CrossEntropy Loss

Correct label: Cat

Multitask Loss

+ → **Loss**

Vector: 4096

Fully Connected: 4096 to 4

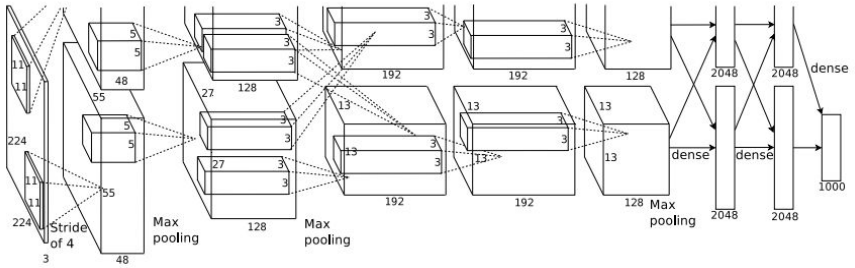
Box Coordinates (x, y, w, h)

L2 Loss

Correct box: (x', y', w', h')

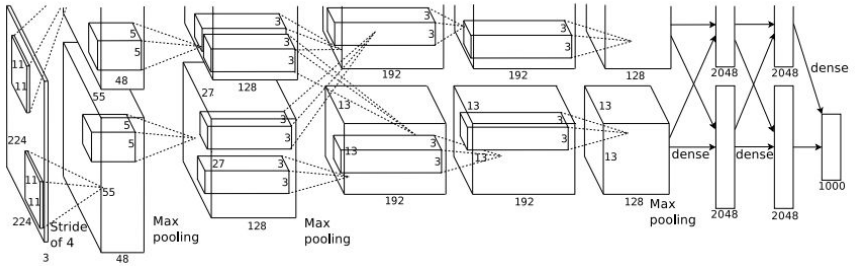
Treat localization as a regression problem!

Object Detection: Multiple Objects

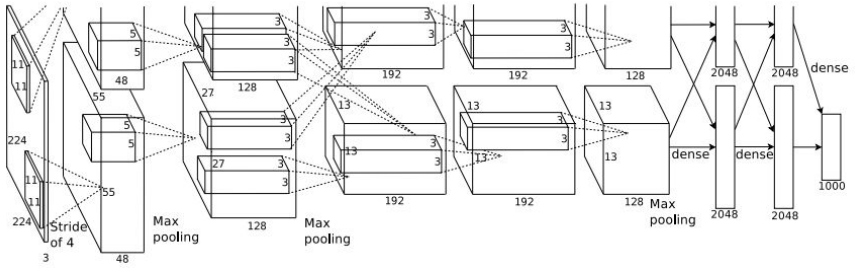


CAT: (x, y, w, h)

Each image needs a different number of outputs!



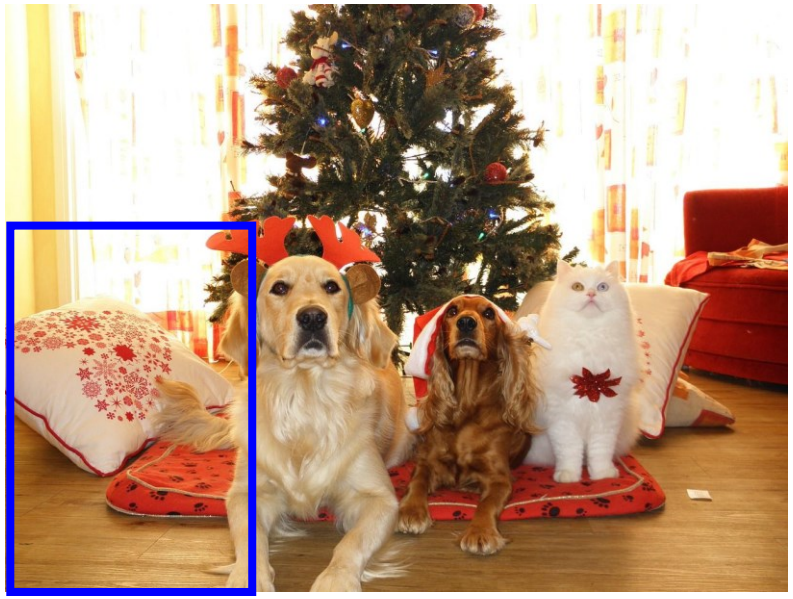
DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)

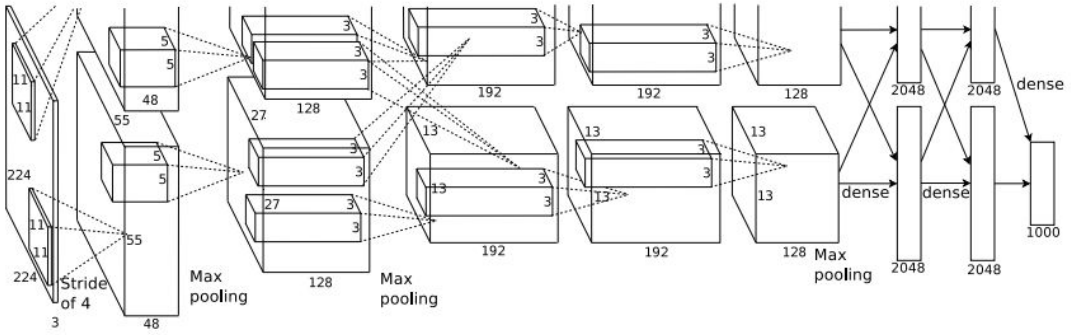


DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
...

Object Detection: Multiple Objects

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

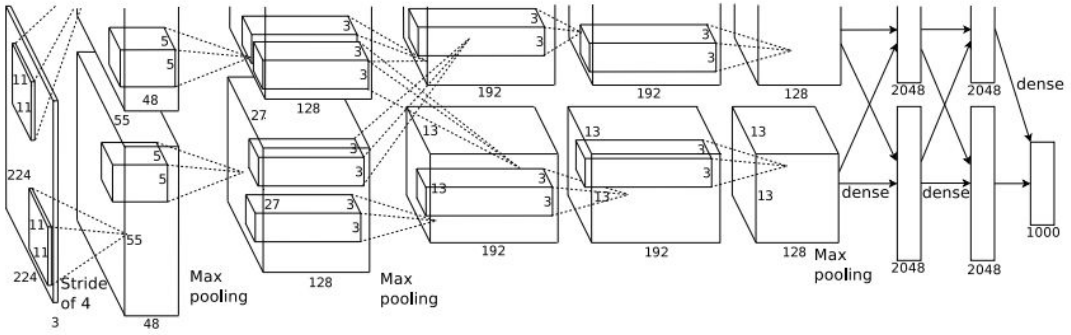




Dog? No
Cat? No
Background? YES

Object Detection: Multiple Objects

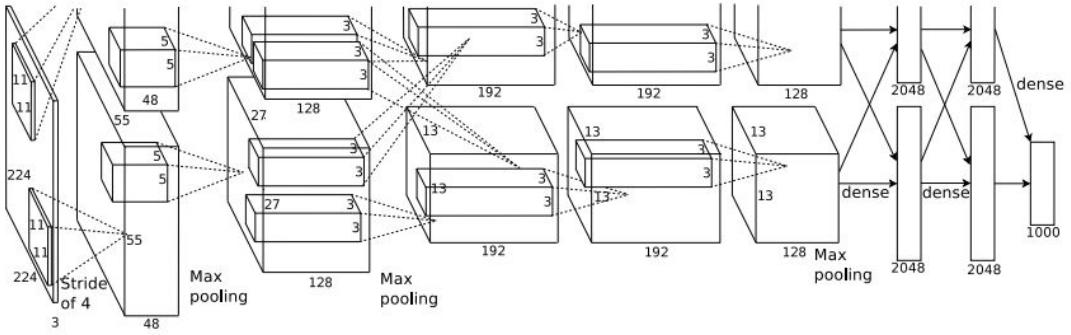
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? Yes
Cat? No
Background? No

Object Detection: Multiple Objects

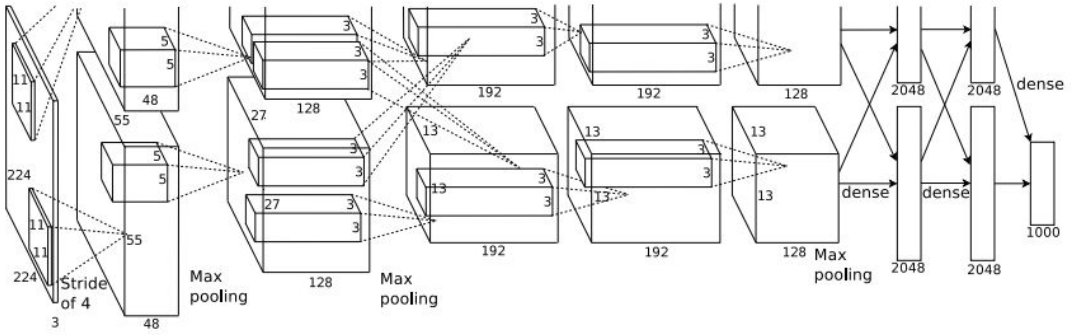
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? Yes
Cat? No
Background? No

Object Detection: Multiple Objects

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

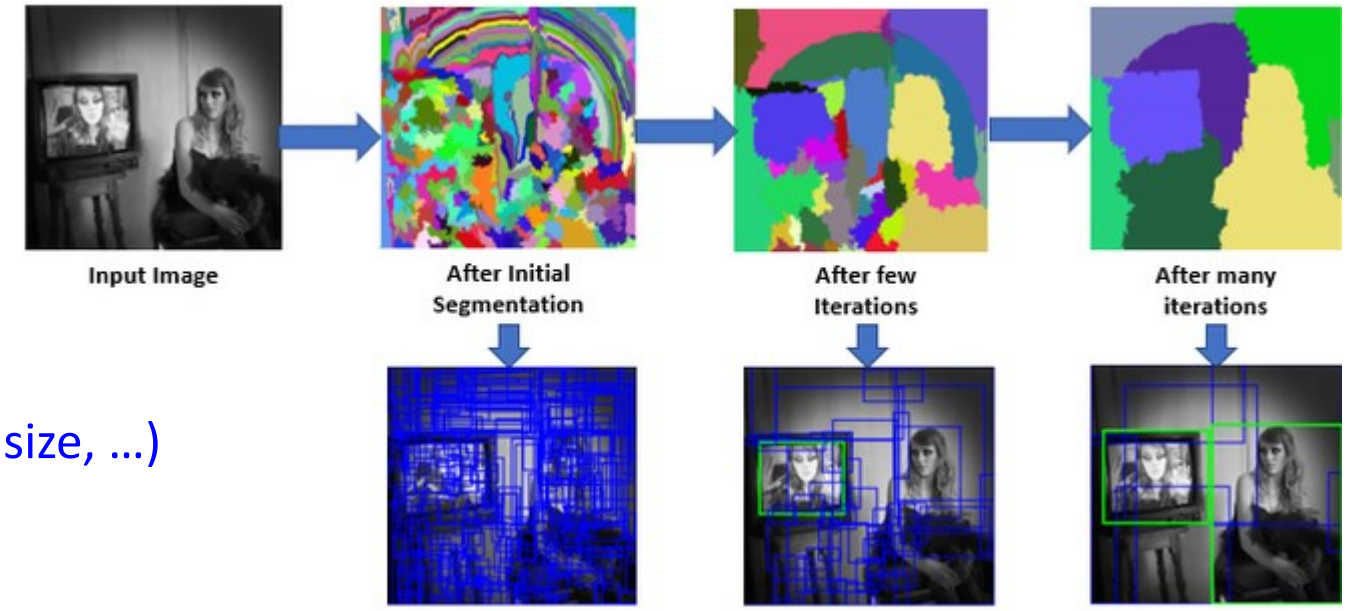


Dog? No
Cat? Yes
Background? No

Problem: Need to apply CNN to huge number of locations, scales, and aspect ratios, very computationally expensive!

Region Proposals: Selective Search

- Find image regions that are likely to contain objects
- Relatively fast to run; e.g. Selective Search gives 2000 region proposals in a few seconds on CPU



Merge the similar region
Similarity: (color, texture, size, …)

Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014.

R-CNN (Region-based Convolutional Neural Net)

Per-image computation

Per-region computation

I:

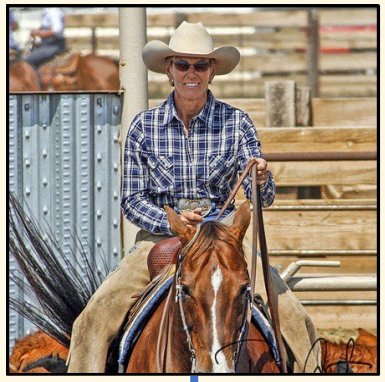
Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014.

R-CNN (Region-based Convolutional Neural Net)

Per-image computation

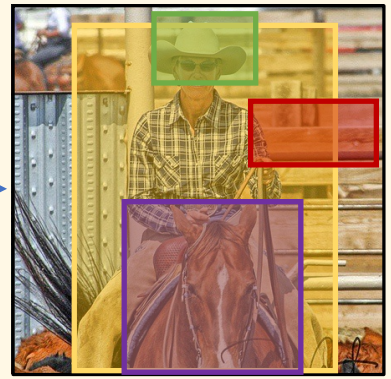
Per-region computation

I:



Selective search, Edge Boxes, MCG, …

1



1. Use an off-the-shelf *Region of Interest* (RoI) proposal algorithm (~2k proposals per image)

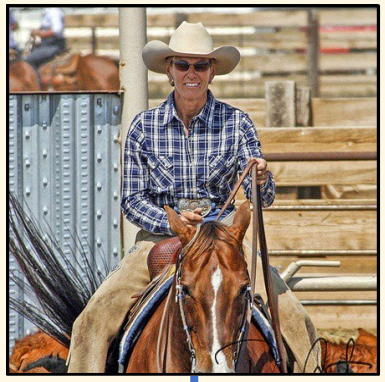
Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014.

R-CNN

Per-image computation

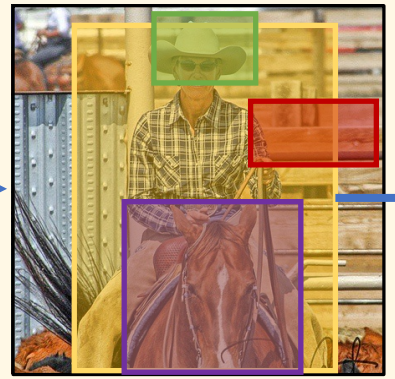
Per-region computation for each $r_i \in r(I)$

I:



Selective search, Edge Boxes, MCG, …

1



Crop & warp

2

2. Crop and warp each proposal image window to obtain a fixed-size network input (224x224)

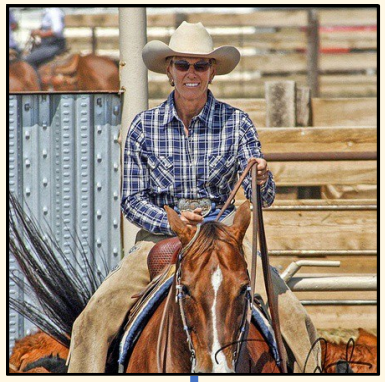
Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014.

R-CNN

Per-image computation

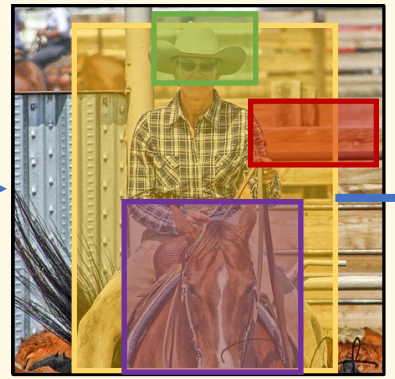
Per-region computation for each $r_i \in r(I)$

I:



Selective search, Edge Boxes, MCG, …

1



Crop & warp

2

ConvNet(r_i)

3

3. Forward propagate the fixed-size network input to get a feature representation (ImageNet-pretranied)

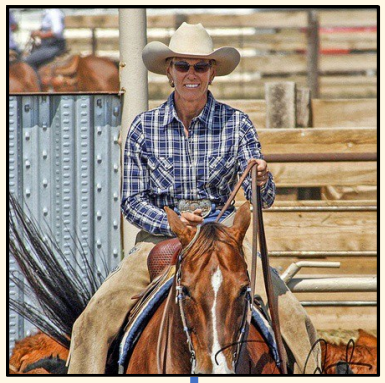
Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014.

R-CNN

Per-image computation

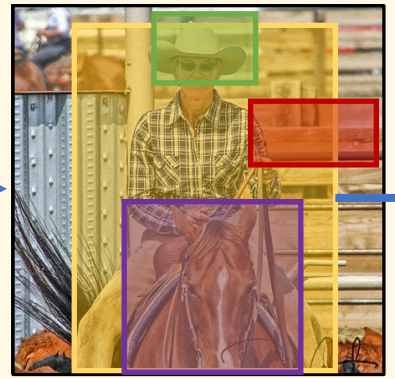
Per-region computation for each $r_i \in r(I)$

I:



Selective search, Edge Boxes, MCG, …

1



Crop & warp

2

ConvNet(r_i)

3

Linear classifier

4

4. Object classification

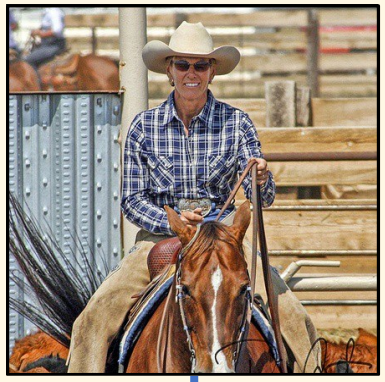
Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014.

R-CNN

Per-image computation

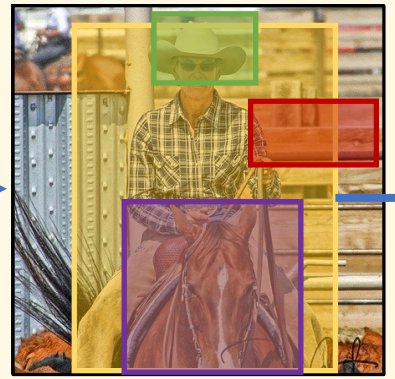
Per-region computation for each $r_i \in r(I)$

I:



Selective search, Edge Boxes, MCG, …

1



Crop & warp

2

ConvNet(r_i)

3

Linear classifier

4

Box regressor

5

5. Refine proposal localization with bounding-box regression

Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014.

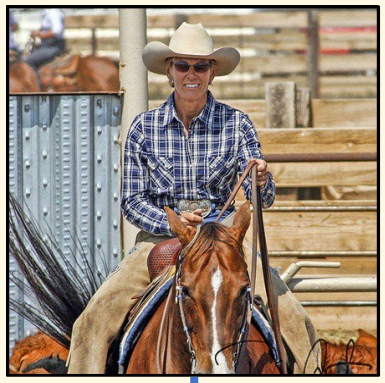
Slow R-CNN

Per-image computation

Per-region computation for each $r_i \in r(I)$

Problem: Very slow! Need to do ~2k independent forward passes for each image!

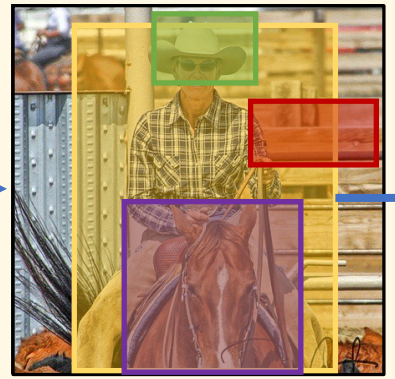
Idea: Pass the image Through convnet before cropping! Crop the conv feature instead!



I:

Selective search, Edge Boxes, MCG, …

1



Crop & warp

2

ConvNet(r_i)

3

Linear classifier

4

Box regressor

5

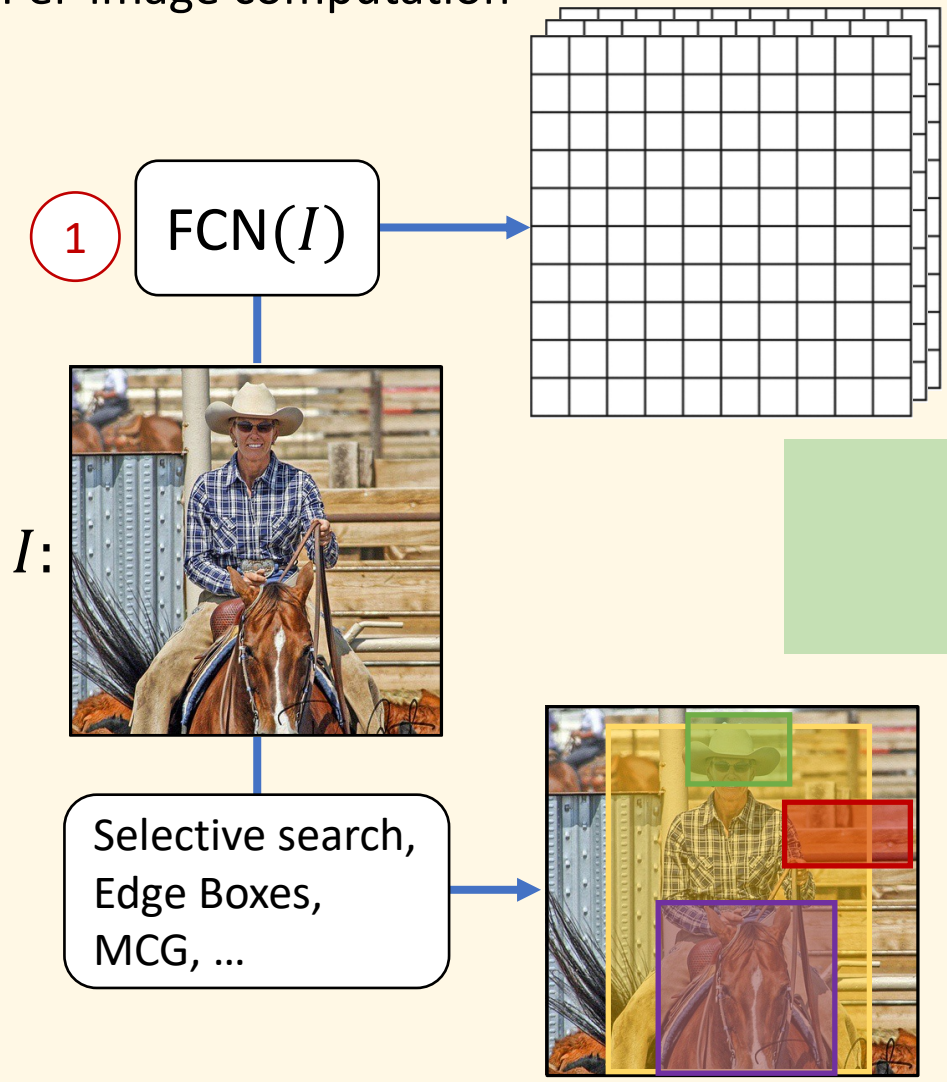
5. Refine proposal localization with bounding-box regression

Girshick. Fast R-CNN. ICCV 2015.

Fast R-CNN

Per-image computation

Per-region computation for each $r_i \in r(I)$



1. Fully convolutional network (FCN) maps the image to a lower resolution spatial feature map

Girshick. Fast R-CNN. ICCV 2015.

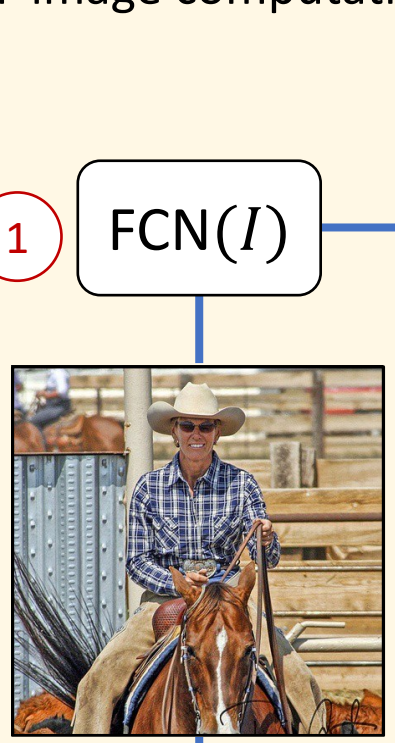
Fast R-CNN

Per-image computation

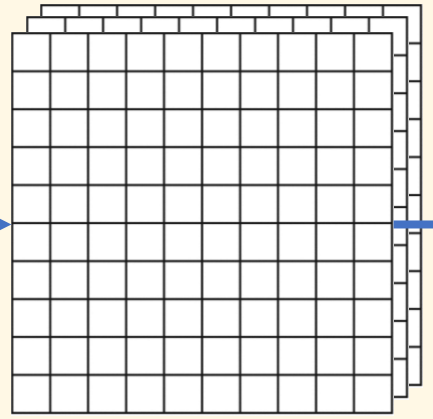
Per-region computation for each $r_i \in r(I)$

1

FCN(I)

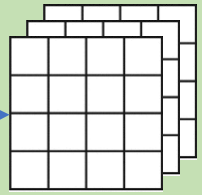


I:

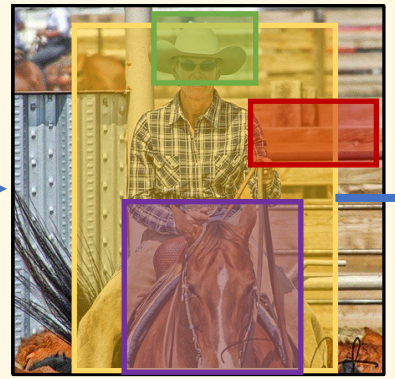


2

RoIPool



Selective search,
Edge Boxes,
MCG, …



2. *Region of interest* (RoI) pooling converts each region into a fixed dimensional representation

Girshick. Fast R-CNN. ICCV 2015.

Fast R-CNN

Per-image computation

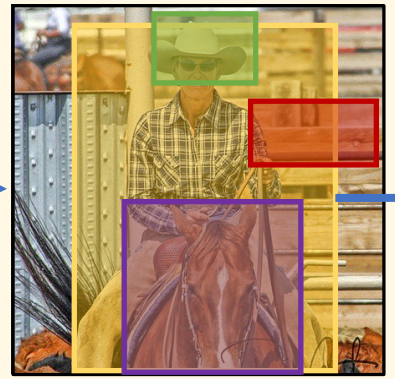
Per-region computation for each $r_i \in r(I)$

1

FCN(I)

I:

Selective search,
Edge Boxes,
MCG, …



2

RoIPool

3

MLP

3. A lightweight MLP processes each region feature map

(Alternatively, the MLP could be a small ConvNet, etc.)

Girshick. Fast R-CNN. ICCV 2015.

Fast R-CNN

Per-image computation

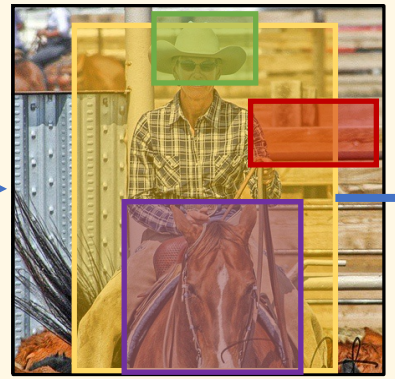
Per-region computation for each $r_i \in r(I)$

1

FCN(I)

I:

Selective search,
Edge Boxes,
MCG, …



2

RoIPool

3

MLP

Softmax clf.

4

Box regressor

4. Classification and box regression heads are applied

Girshick. Fast R-CNN. ICCV 2015.

Fast R-CNN

Per-image computation

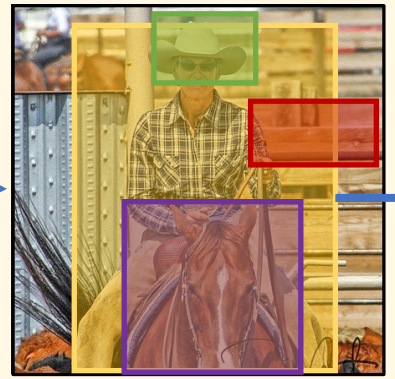
Per-region computation for each $r_i \in r(I)$

1

FCN(I)

I:

Selective search,
Edge Boxes,
MCG, …



2

RoIPool

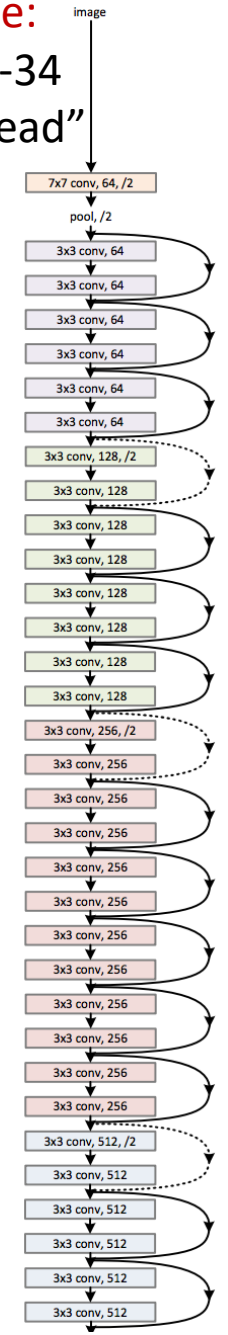
MLP

Softmax clf.

Box regressor

Whole-image, Fully Conv. Network (FCN)

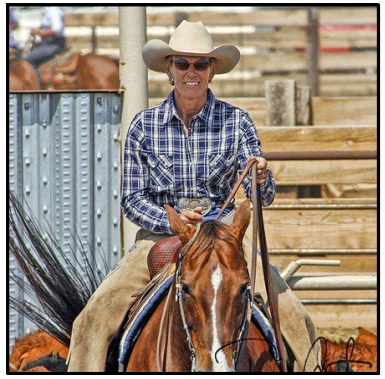
Example:
ResNet-34
w/o “head”



Use any standard ConvNet as the “backbone architecture”

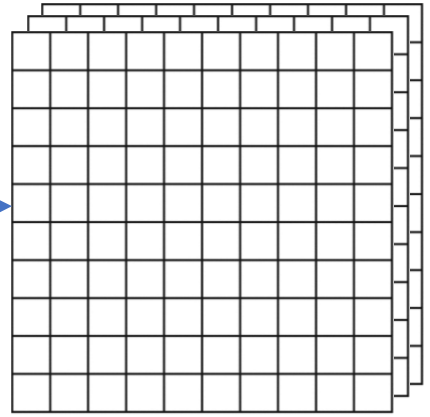
- AlexNet, VGG, ResNet, Inception, Inception-ResNet, ResNeXt, DenseNet, NAS*, …
- Remove global pooling / FC
- Output spatial dims are proportional to input spatial dims

I: H



W

FCN(I)



Example feature map dimensions:
(512, H/16, W/16)

The Engine of Recognition

A good network lift all boats (“features matter”)

- AlexNet
- VGG
- GoogleNet / Inception
- ResNet
- ResNeXt
- NAS-derived networks
- …

Girshick. Fast R-CNN. ICCV 2015.

Fast R-CNN

Per-image computation

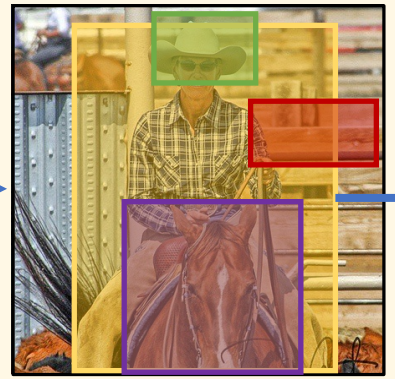
Per-region computation for each $r_i \in r(I)$

1

FCN(I)

I:

Selective search,
Edge Boxes,
MCG, …



2

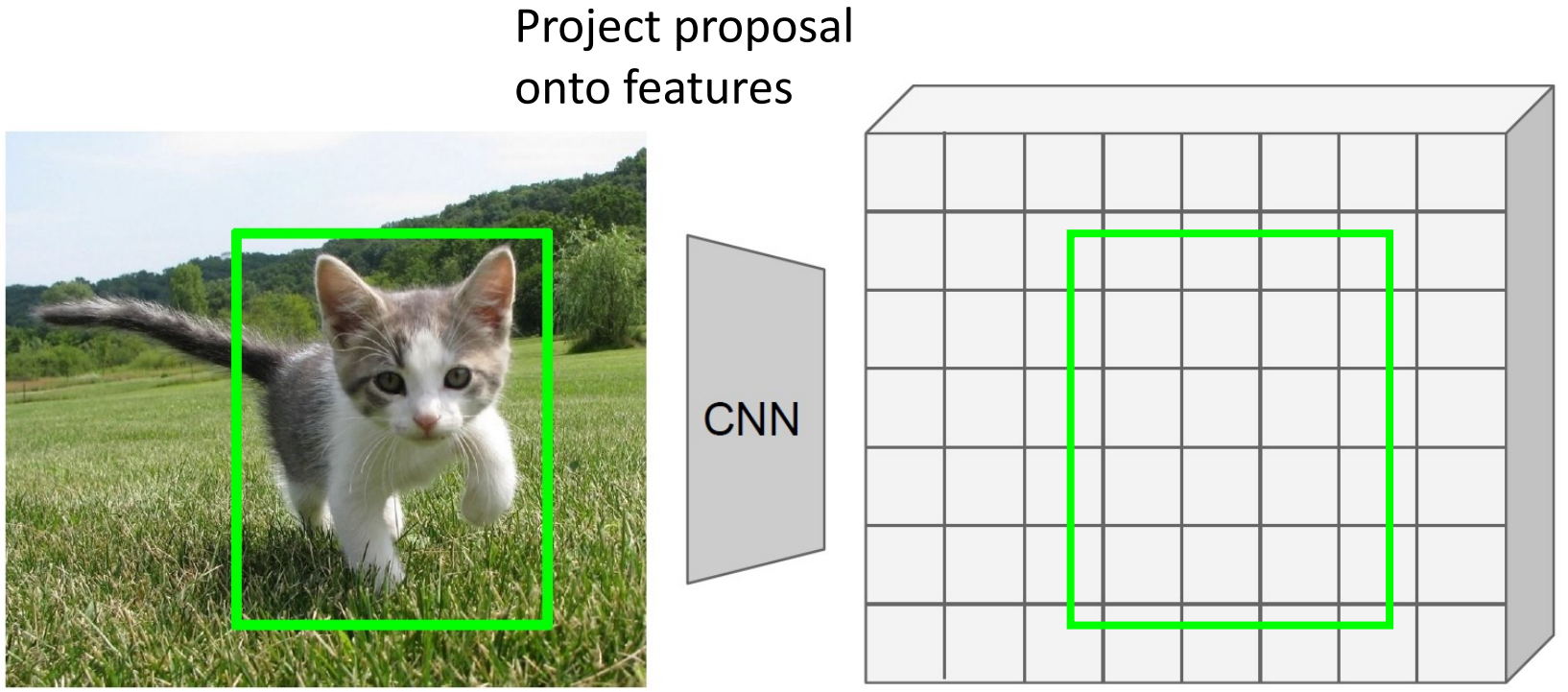
RoIPool

MLP

Softmax clf.

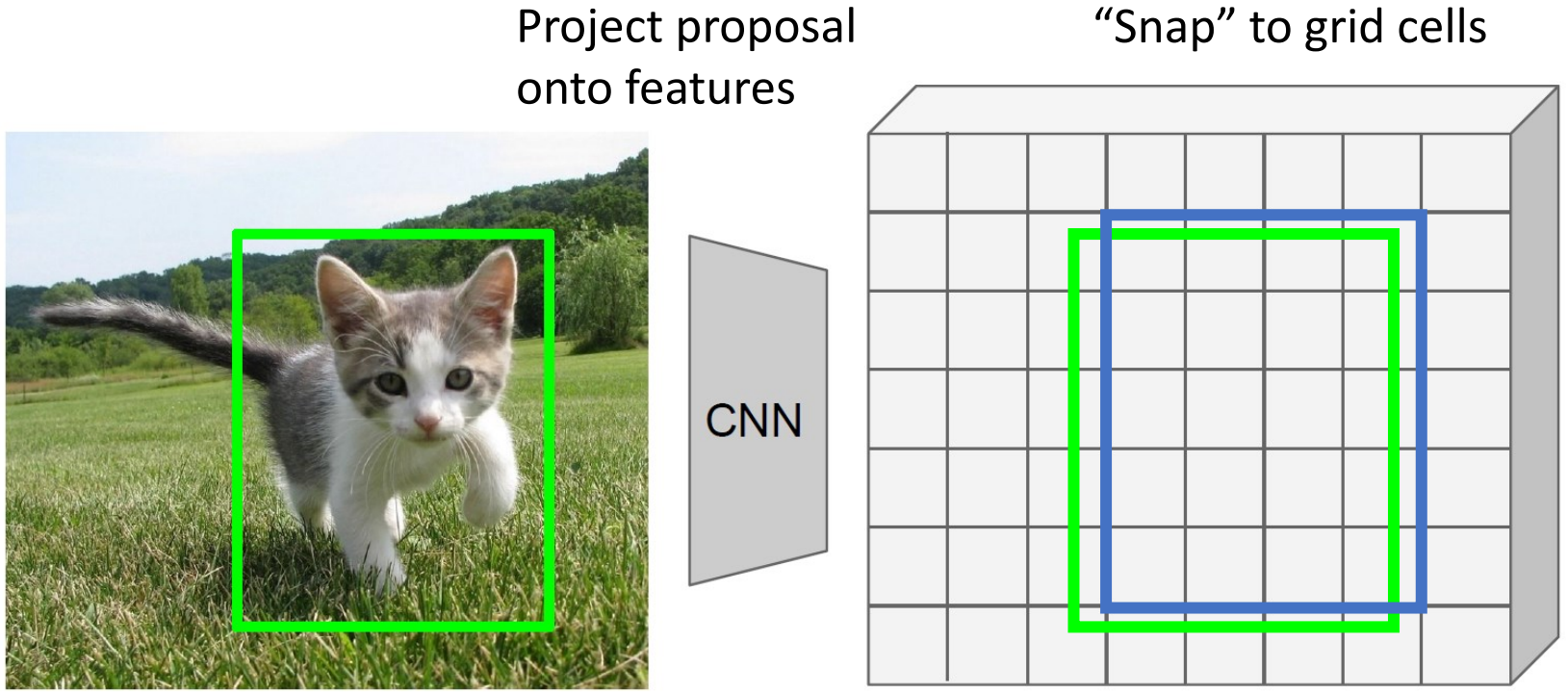
Box regressor

Cropping Features: RoIPool



Girshick. Fast R-CNN. ICCV 2015.

Cropping Features: RoIPool

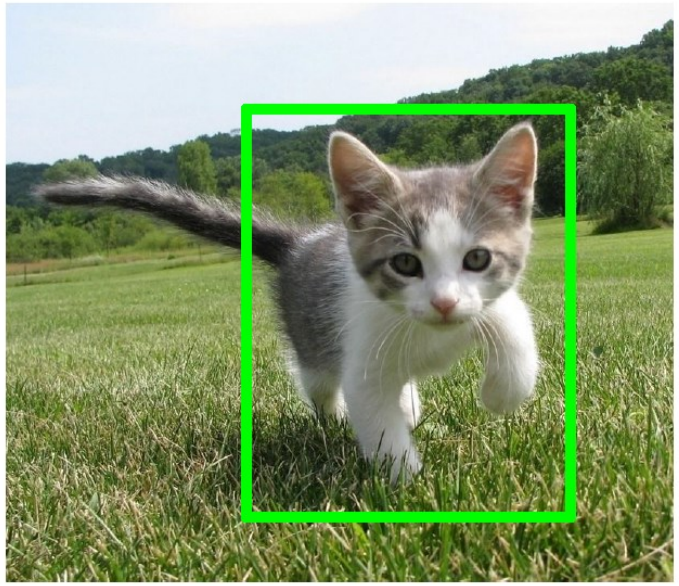


Divide into 2x2 grid of (roughly) equal subregions

Q: how do we resize the 512 mx 5 x 4 region to, e.g., a 512 x 2 x 2 tensor?

Girshick. Fast R-CNN. ICCV 2015.

Cropping Features: RoIPool



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

Project proposal onto features

CNN

“Snap” to grid cells

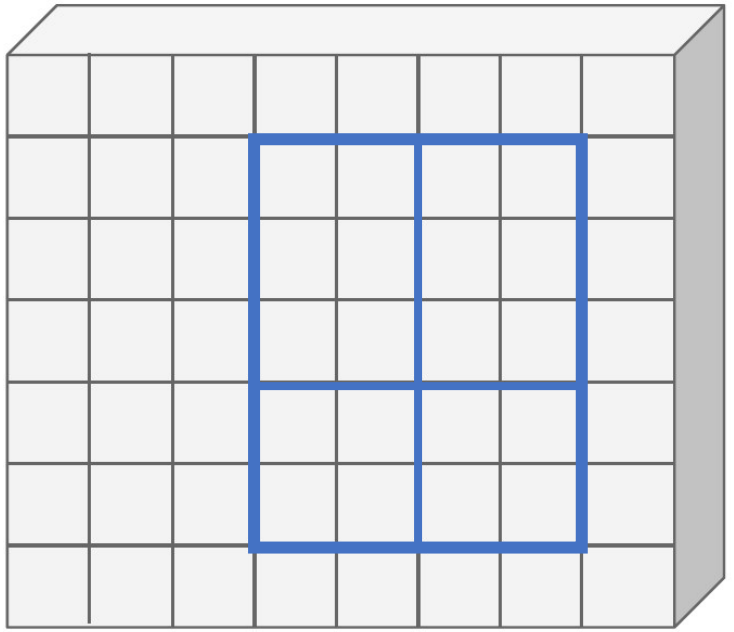
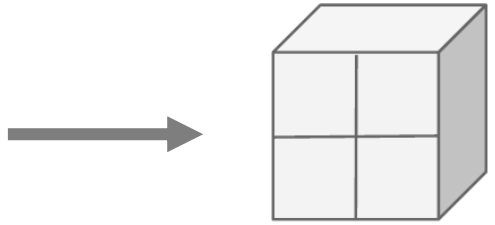


Image features: C x H x W
(e.g. 512 x 20 x 15)

Divide into 2x2 grid of (roughly) equal subregions

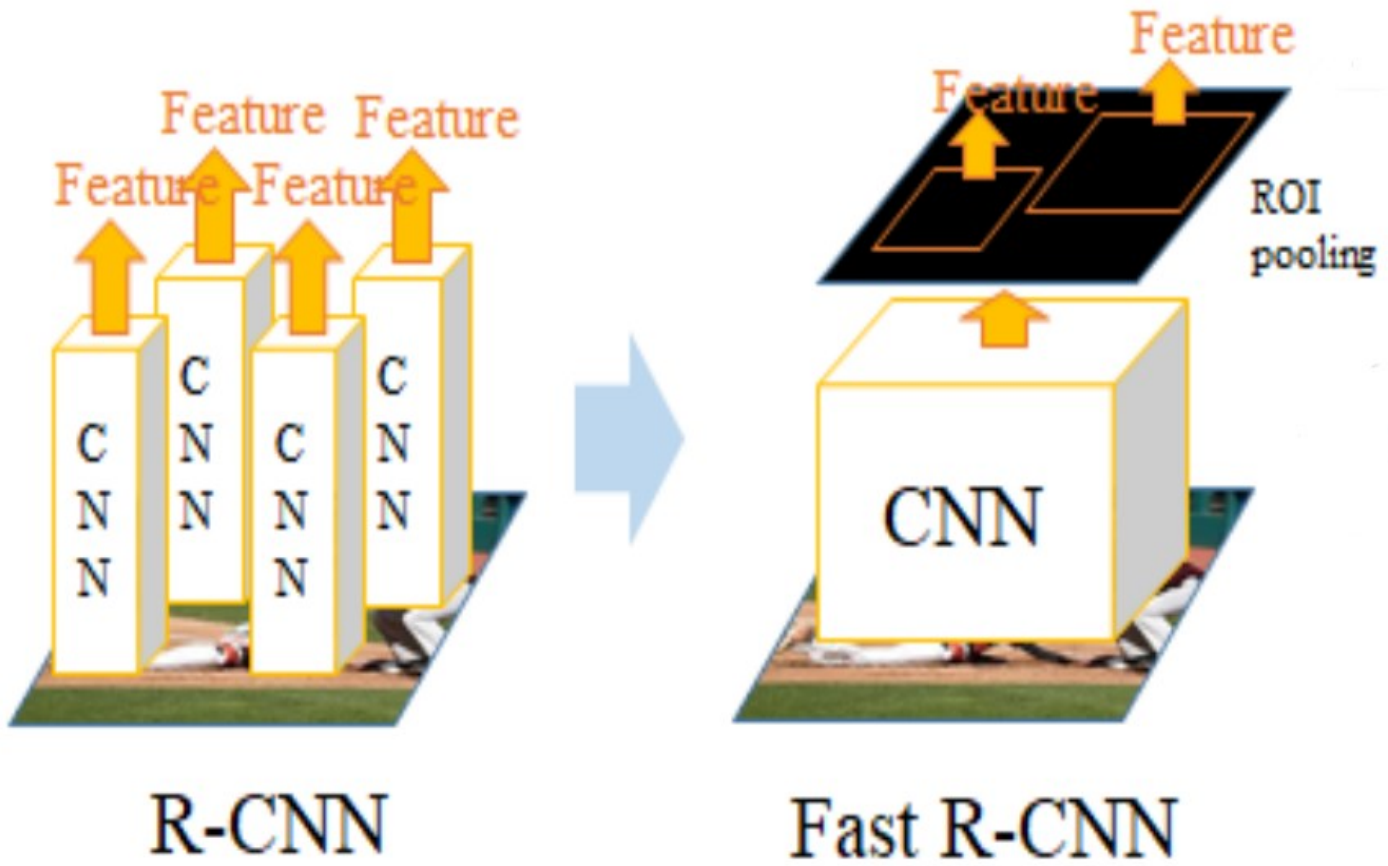
Max-pool within each subregion



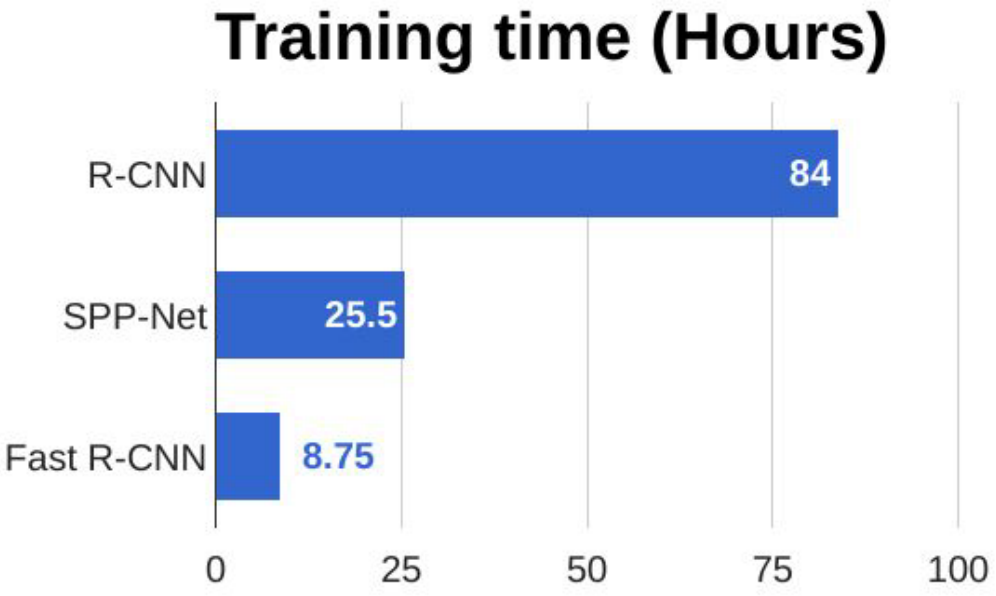
Region features
(here 512 x 2 x 2;
In practice e.g 512 x 7 x 7)

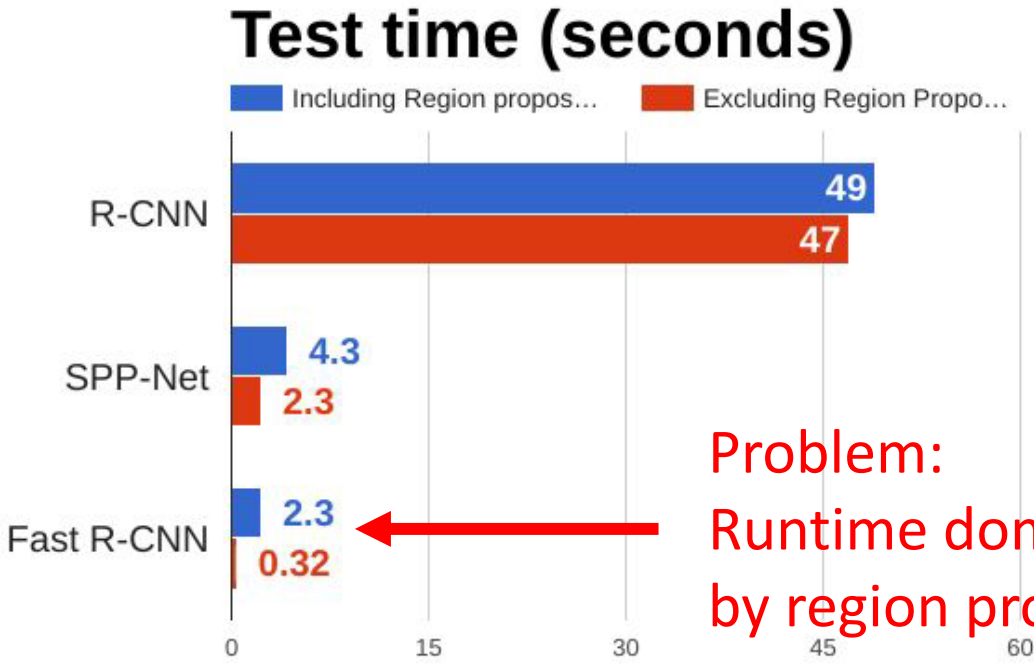
Region features always the same size even if input regions have different sizes!

R-CNN vs Fast R-CNN



R-CNN vs Fast R-CNN





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

Ren, He, Girshick, Sun. Faster R-CNN:
Towards Real-Time Object Detection. NIPS 2015.

Faster R-CNN

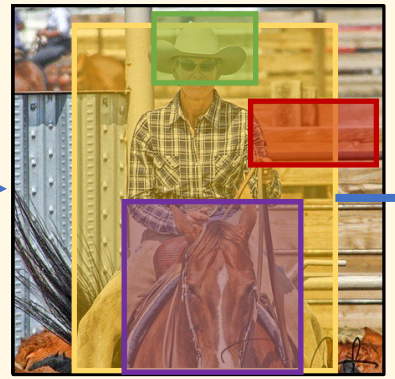
Per-image computation

Per-region computation for each $r_i \in r(I)$

f_I = FCN(I)

I:

RPN(f_I)



RoIPool

MLP

Softmax clf.

Box regressor

Learned proposals
Shares computation with whole-image network

Ren, He, Girshick, Sun. Faster R-CNN:
Towards Real-Time Object Detection. NIPS 2015.

Faster R-CNN

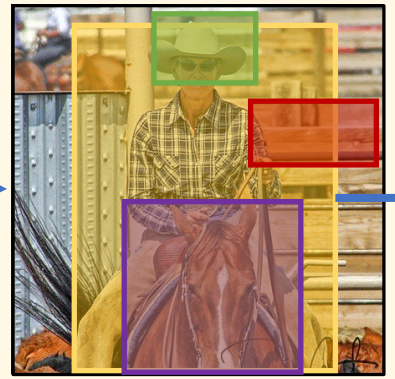
Per-image computation

Per-region computation for each $r_i \in r(I)$

f_I = FCN(I)

I:

RPN(f_I)



RoIPool

MLP

Softmax clf.

Box regressor

Learned proposals
Shares computation with whole-image network

RPN: Region Proposal Network

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

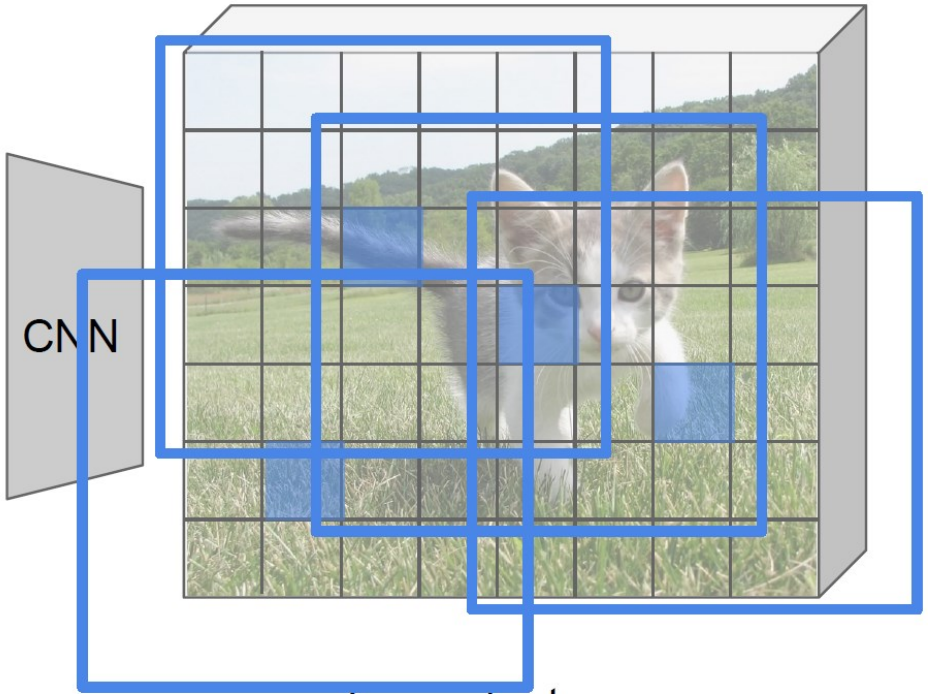
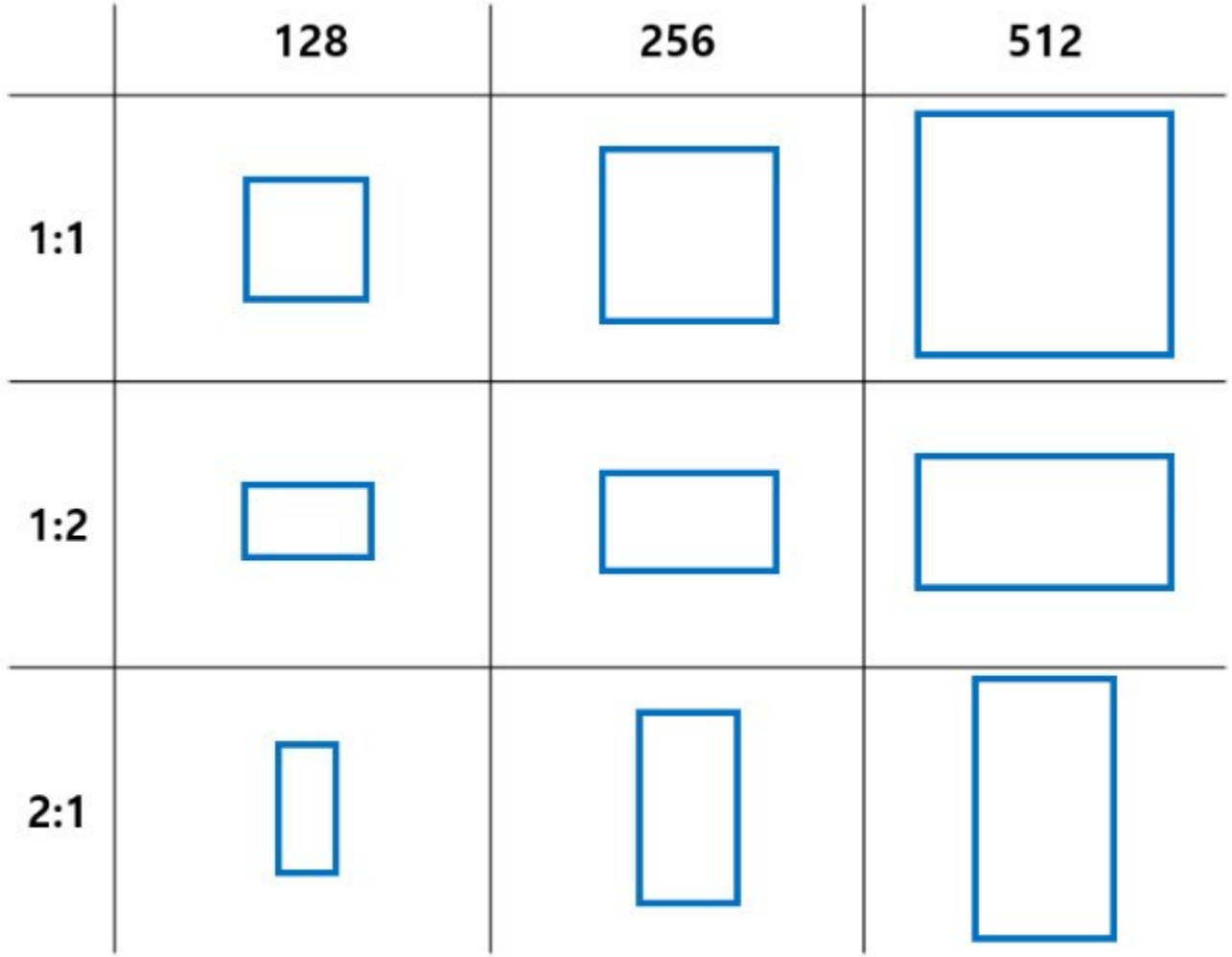
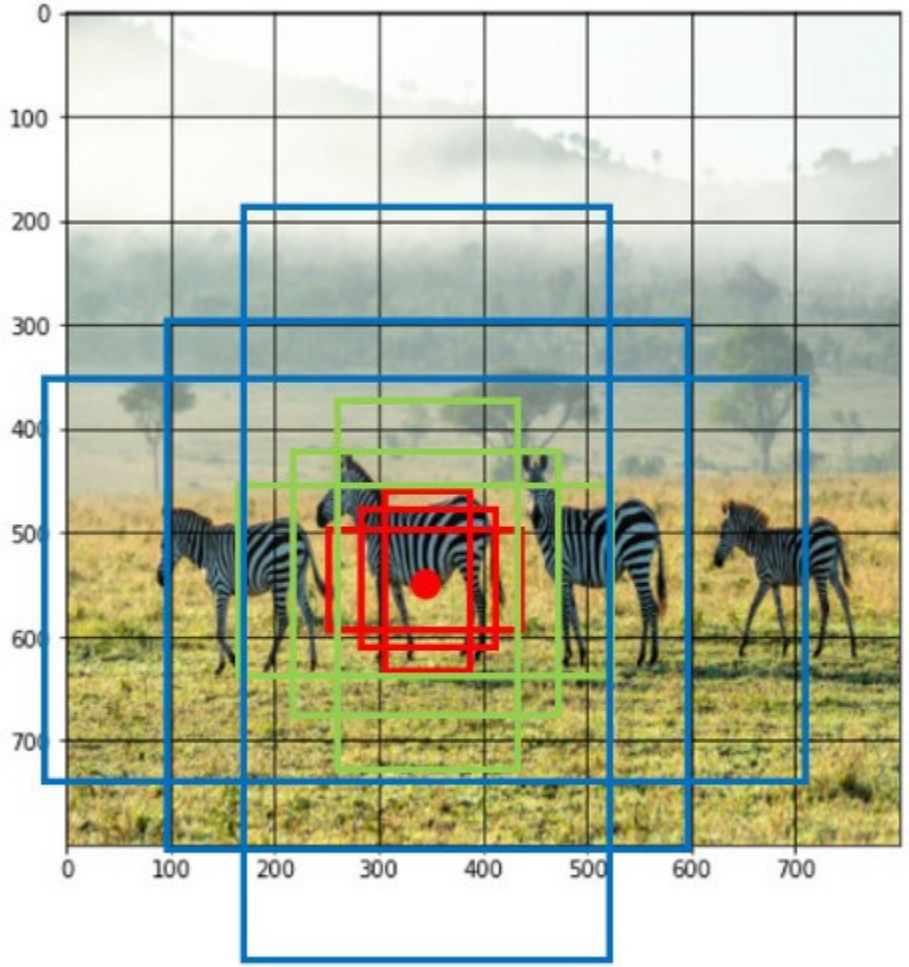


Image features: C x H x W
(e.g. 512 x 20 x 15)

RPN: Region Proposal Network





RPN: Region Proposal Network

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

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

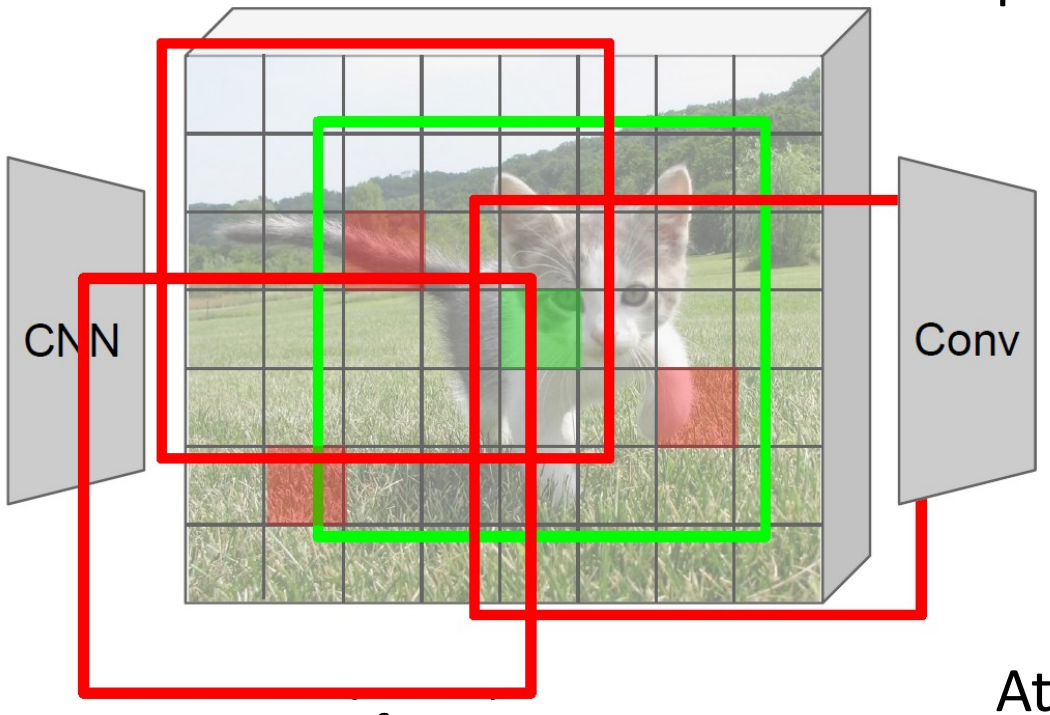


Image features: C x H x W
(e.g. 512 x 20 x 15)

Anchor is an object?
2 x 9 x 20 x 15

At each point, predict whether the corresponding anchor contains an object (binary classification)

Intersection over Union

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

IoU of ground truth box and anchor box

Highest IoU	Positive box
IoU > 0.7	Positive box
IoU < 0.3	Negative box
Others	do not use in training

RPN: Region Proposal Network

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

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

CNN

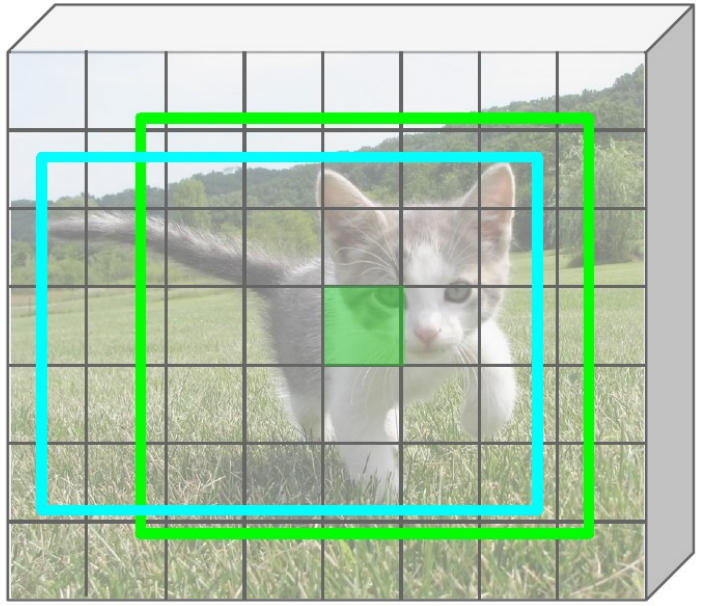


Image features: C x H x W
(e.g. 512 x 20 x 15)

Conv

Anchor is an object?
2 x 9 x 20 x 15

Box corrections
4 x 9 x 20 x 15

For positive boxes, also predict a corrections from the anchor to the ground-truth box (regress 4 numbers per pixel)

RPN: Region Proposal Network

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

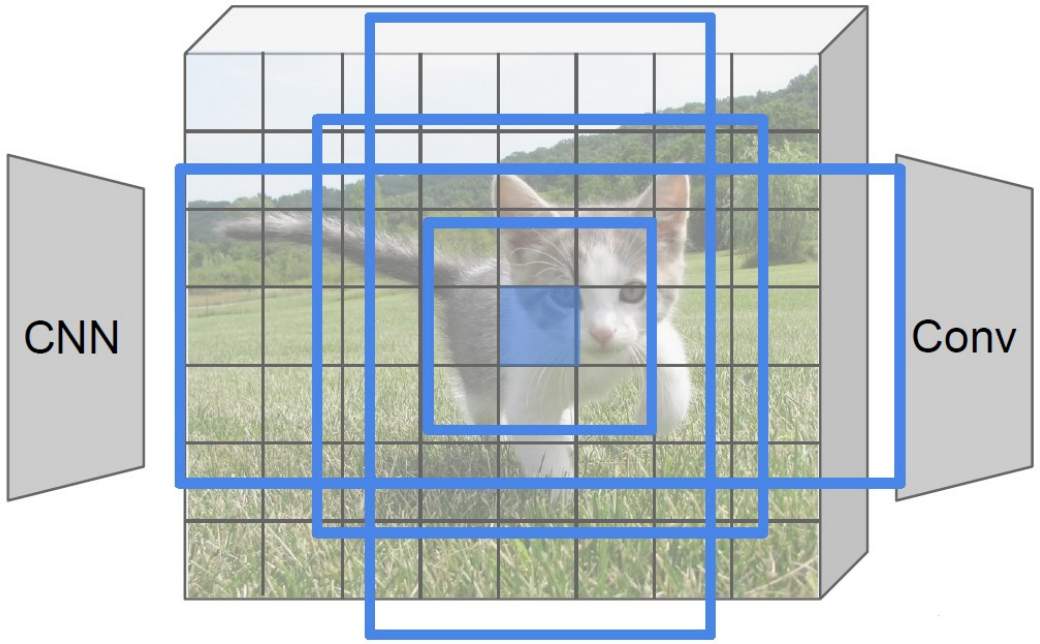


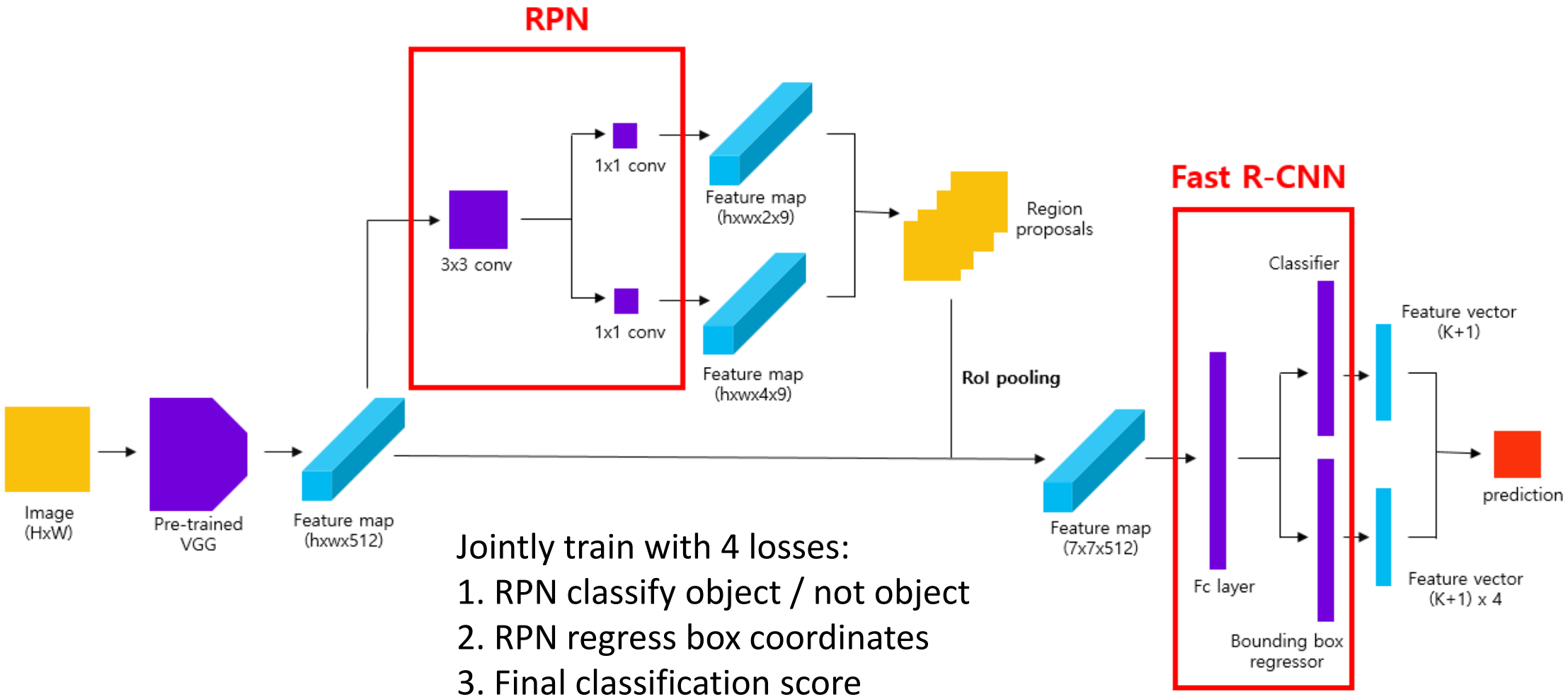
Image features: C x H x W
(e.g. 512 x 20 x 15)

Anchor is an object?
2 x 9 x 20 x 15

Box corrections
4 x 9 x 20 x 15

Sort the 9*20*15 boxes by their “objectness” score, take top ~300 as our proposals

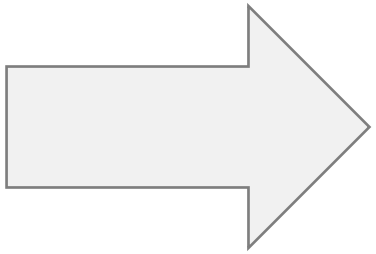
RPN: Region Proposal Network



Jointly train with 4 losses:

1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score
4. Final box coordinates

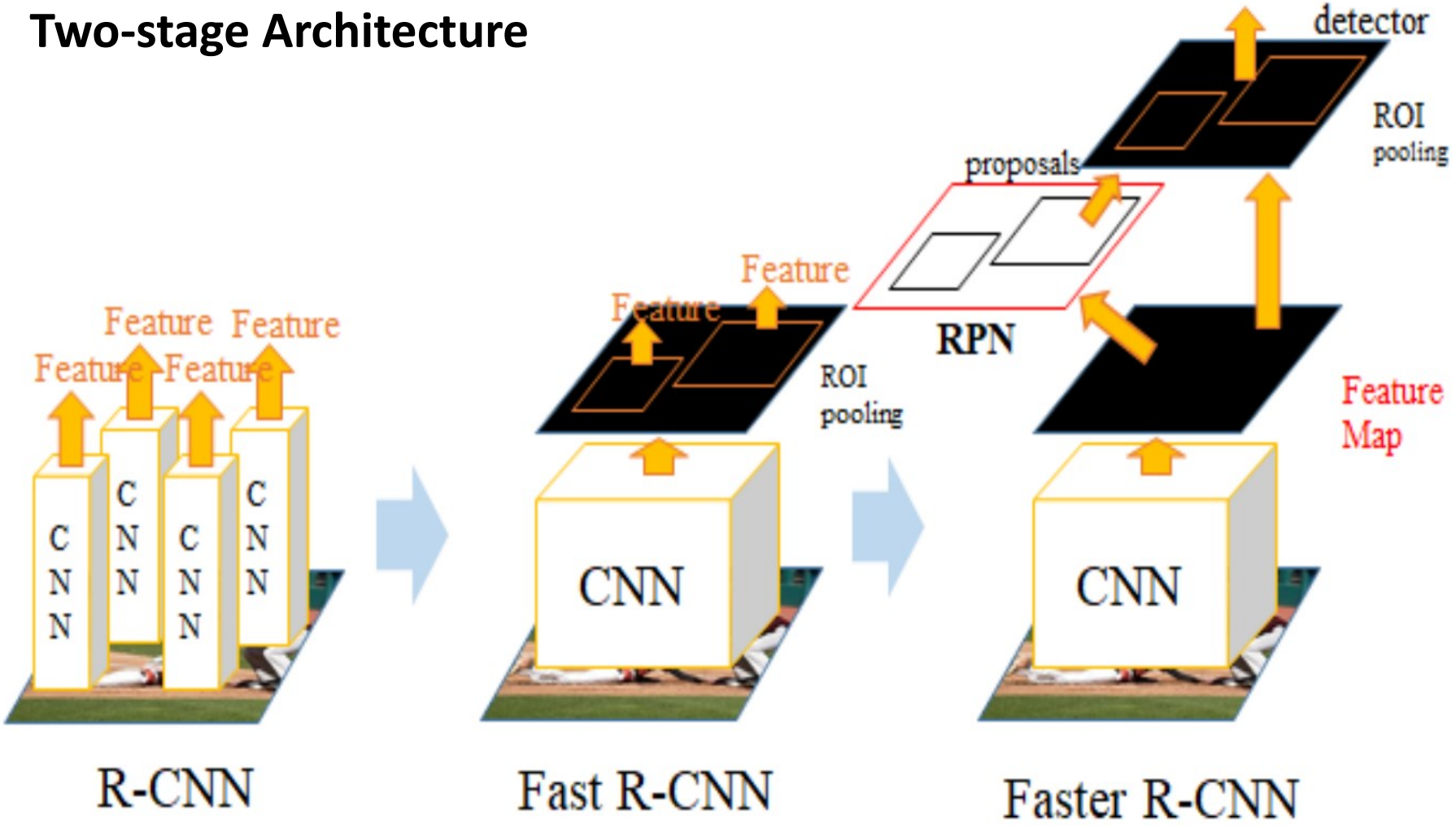
Non-Maximum Suppression



If IoU > 0.7, remove the box with lower confidence

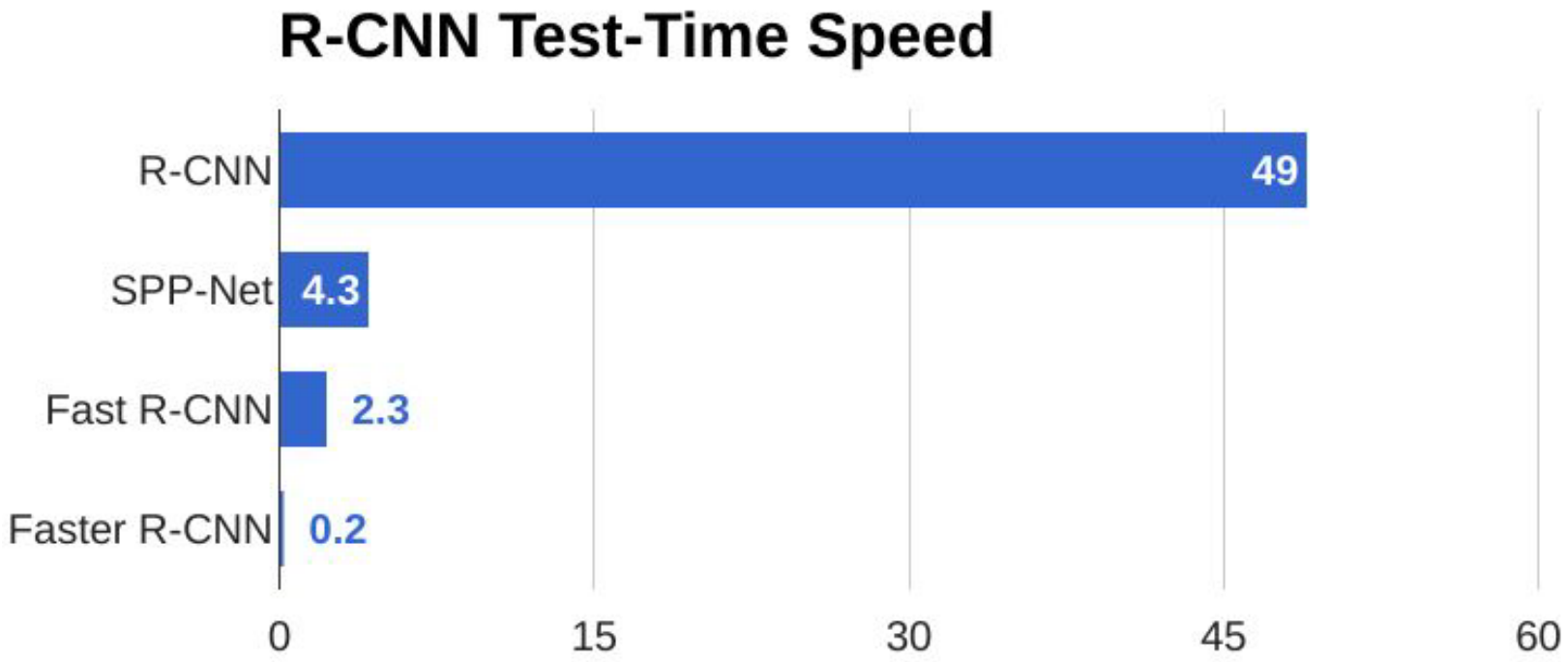
R-CNN vs Fast R-CNN vs Faster R-CNN

Two-stage Architecture



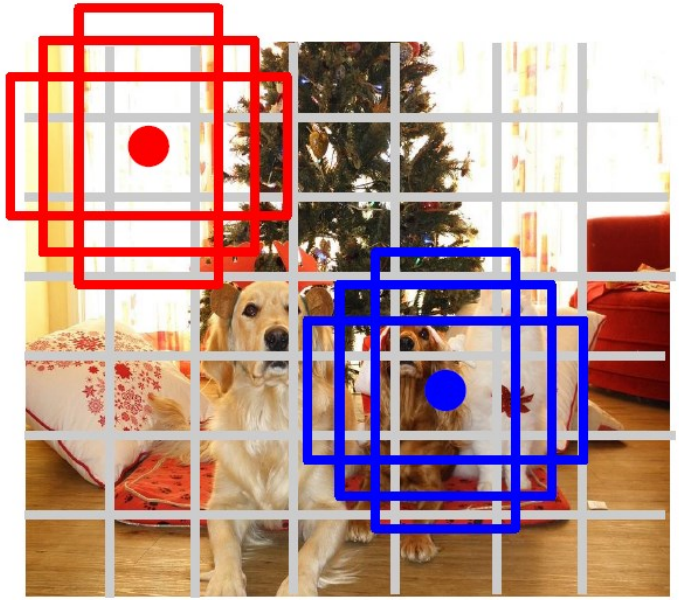
Faster R-CNN

- Make CNN do proposal!



Single-Stage Object Detectors: YOLO

Input image
3 x H x W



Divide image into grid
512 x 7 x 7

Image a set of base boxes centered at each grid cell
Here B = 3

Within each grid cell:

- Regress from each of the **B** base boxes to a final box with 5 numbers:
  (dx, dy, dh, dw, confidence)
- Predict scores for each of **C** classes (including background as a class)
- Looks a lot like RPN, but category-specific!

Output:
7 x 7 x (5 * B + C)

Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016

Ren, He, Girshick, Sun. Faster R-CNN:
Towards Real-Time Object Detection. NIPS 2015.

Faster R-CNN

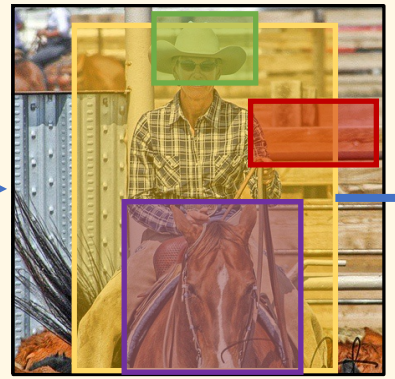
Per-image computation

Per-region computation for each $r_i \in r(I)$

f_I = FCN(I)

I:

RPN(f_I)



RoIPool

MLP

Softmax clf.

Box regressor

Learned proposals
Shares computation with whole-image network

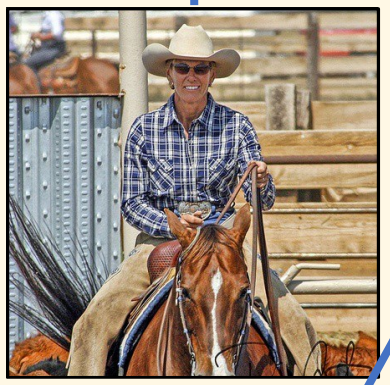
Faster R-CNN with a Feature Pyramid Network

Per-image computation

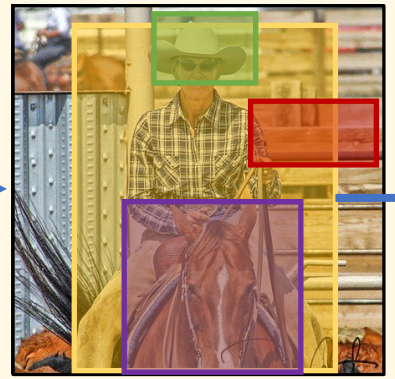
Per-region computation for each $r_i \in r(I)$

f_I = FPN(I)

I:



RPN(f_I)



RoIPool

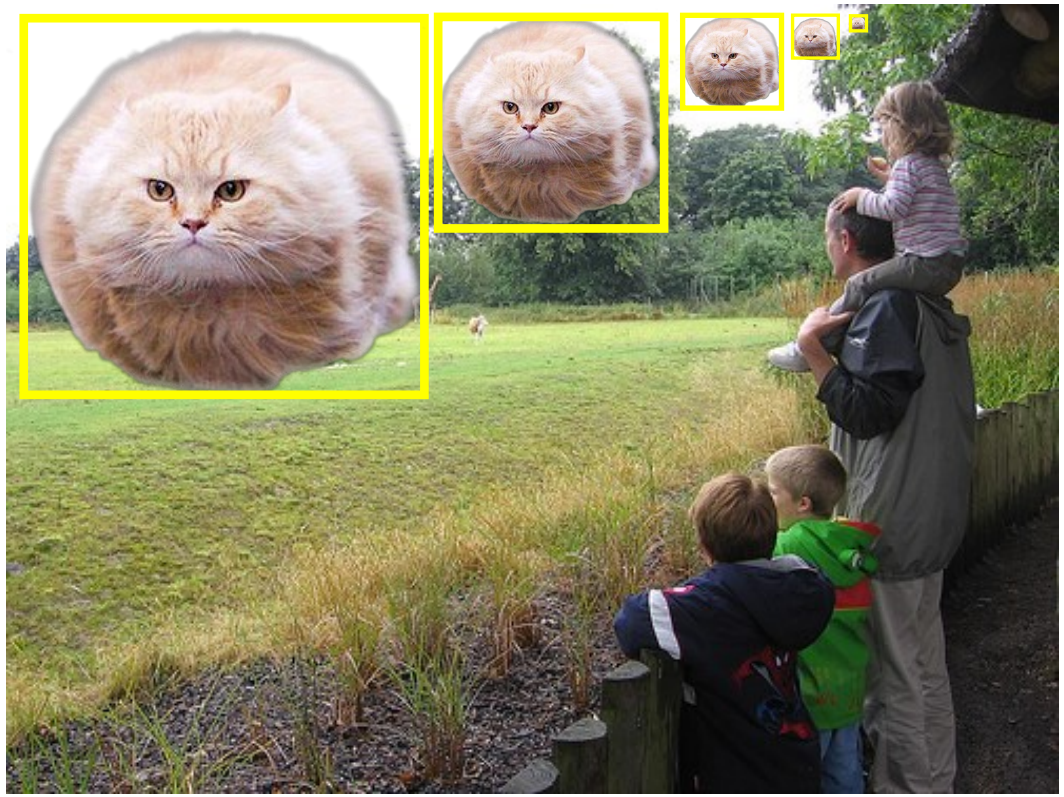
MLP

Softmax clf.

Box regressor

The whole-image feature representation can be improved by making it *multi-scale*

FPN: Improving Scale Invariance and Equivariance



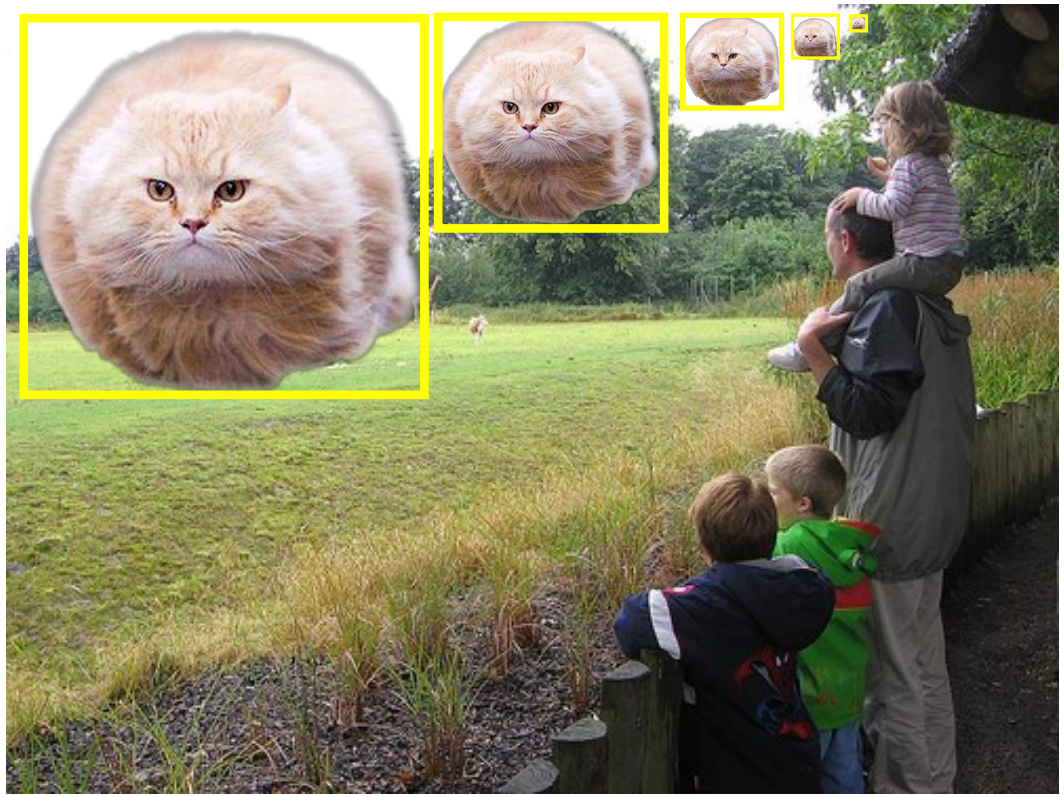
Detectors need to

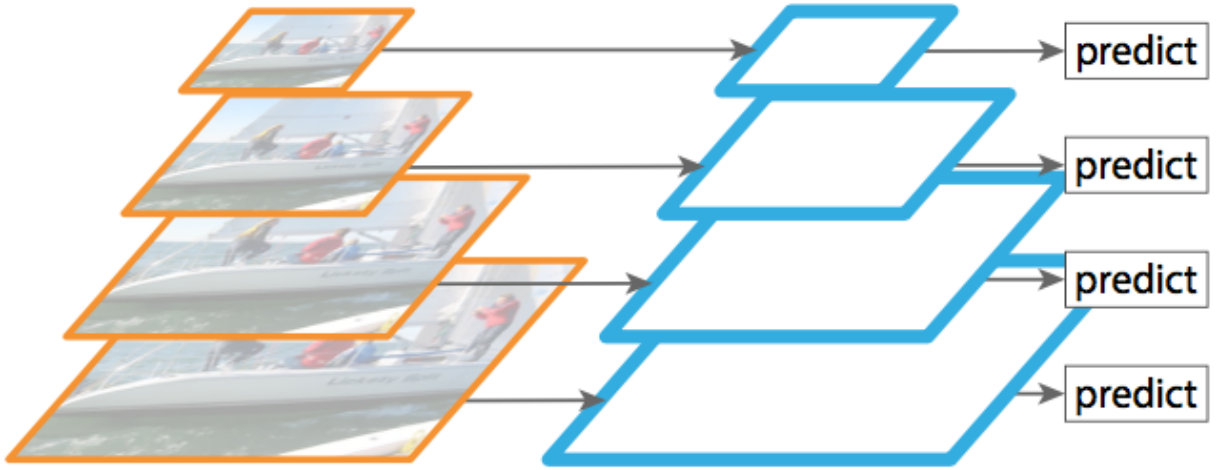
1. Classify (invariance) and
2. Localize (equivariance)

objects over a wide range of scales

FPN improves this ability

Strategy 1: Image Pyramid



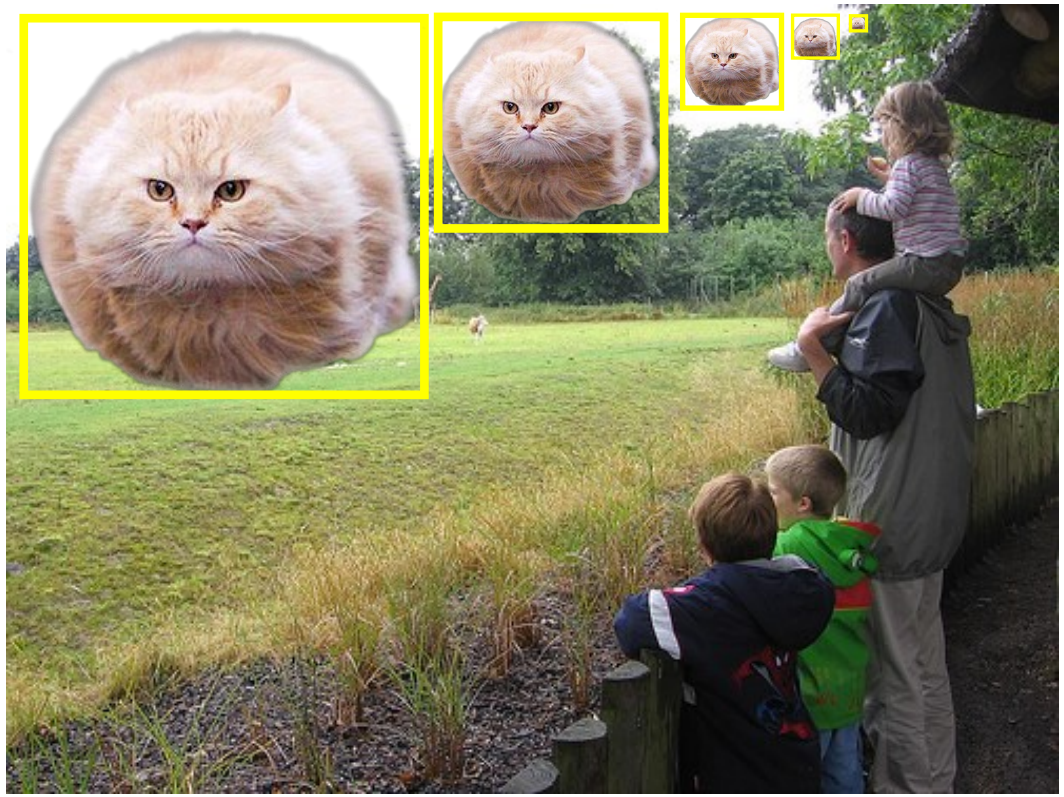


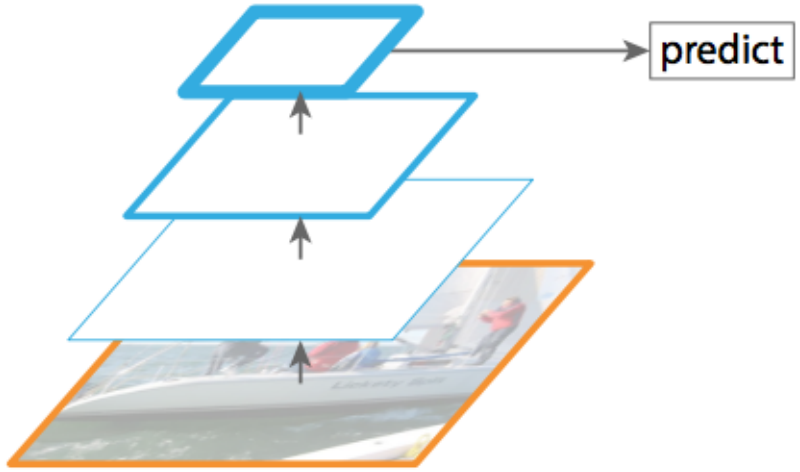
(a) Featurized image pyramid

Standard solution – *slow!*

(E.g., Viola & Jones, HOG, DPM, SPP-net, multi-scale Fast R-CNN, …)

Strategy 2: Multi-scale Features (Single-scale Map)



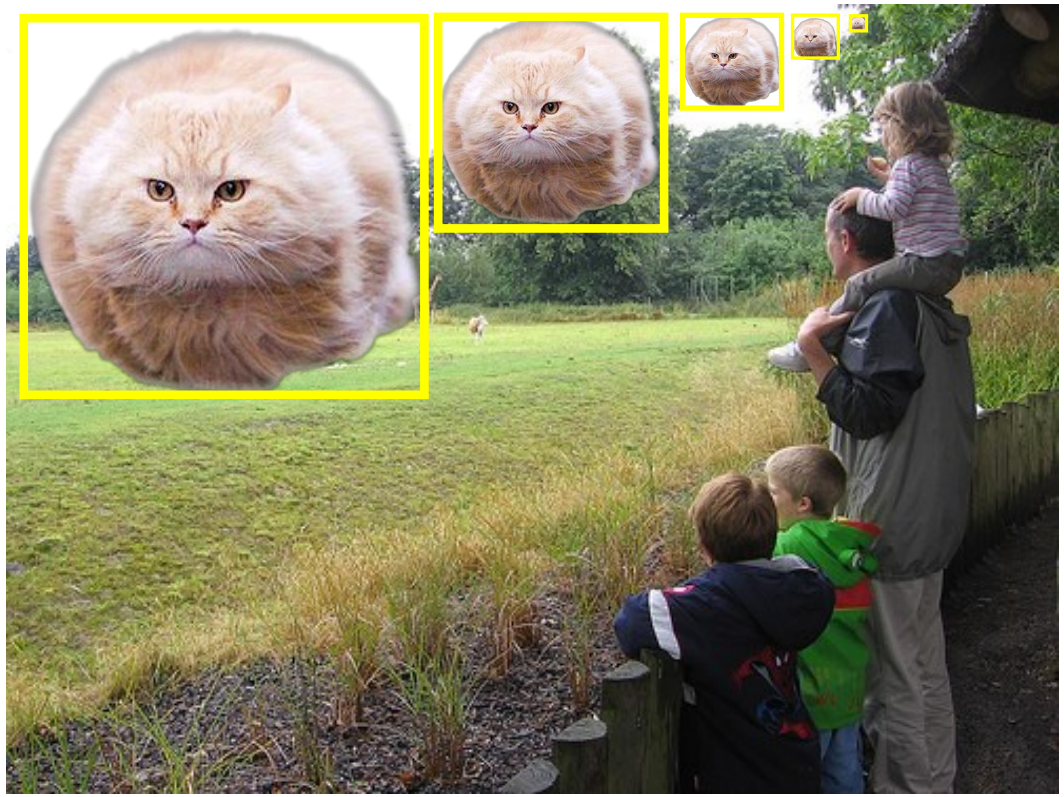


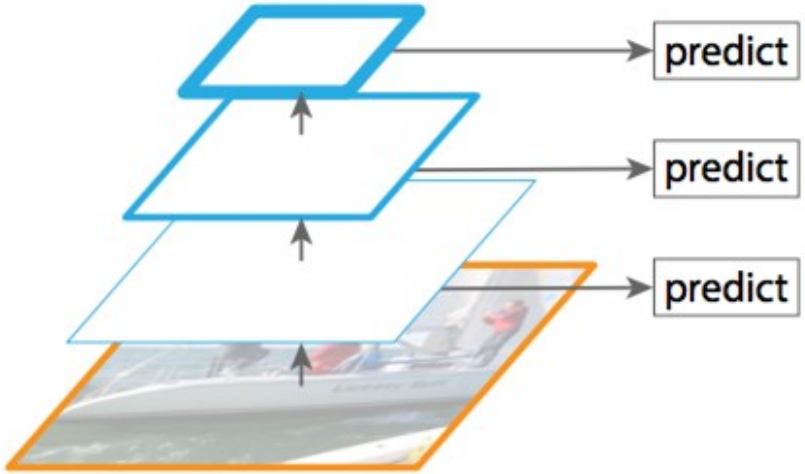
(b) Single feature map

Leave it all to the features – *fast, suboptimal*

(E.g., Fast/er R-CNN, YOLO, …)

Strategy 3: Naïve In-network Pyramid



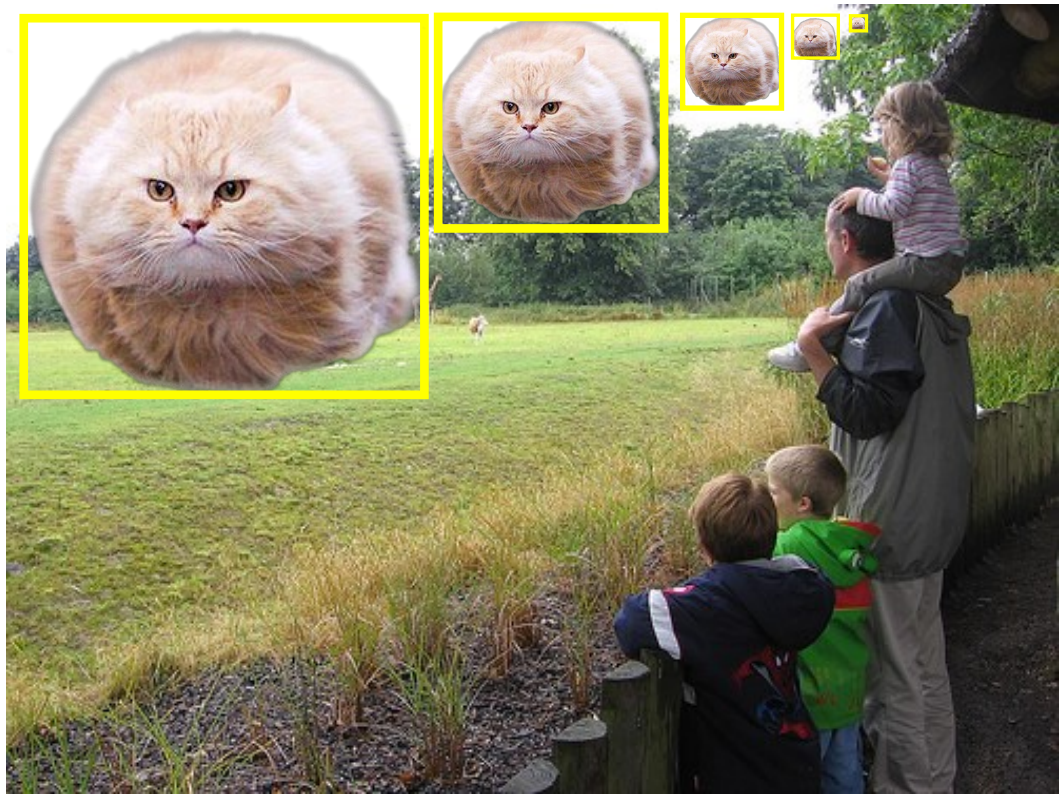


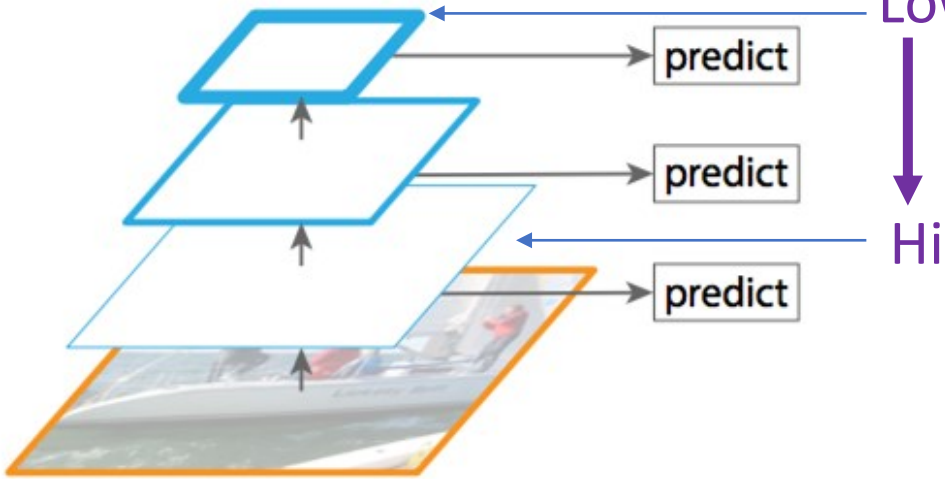
(c) Pyramidal feature hierarchy

Use the internal pyramid – *fast, suboptimal*

(E.g., ≈ SSD, …)

Strategy 3: Naïve In-network Pyramid



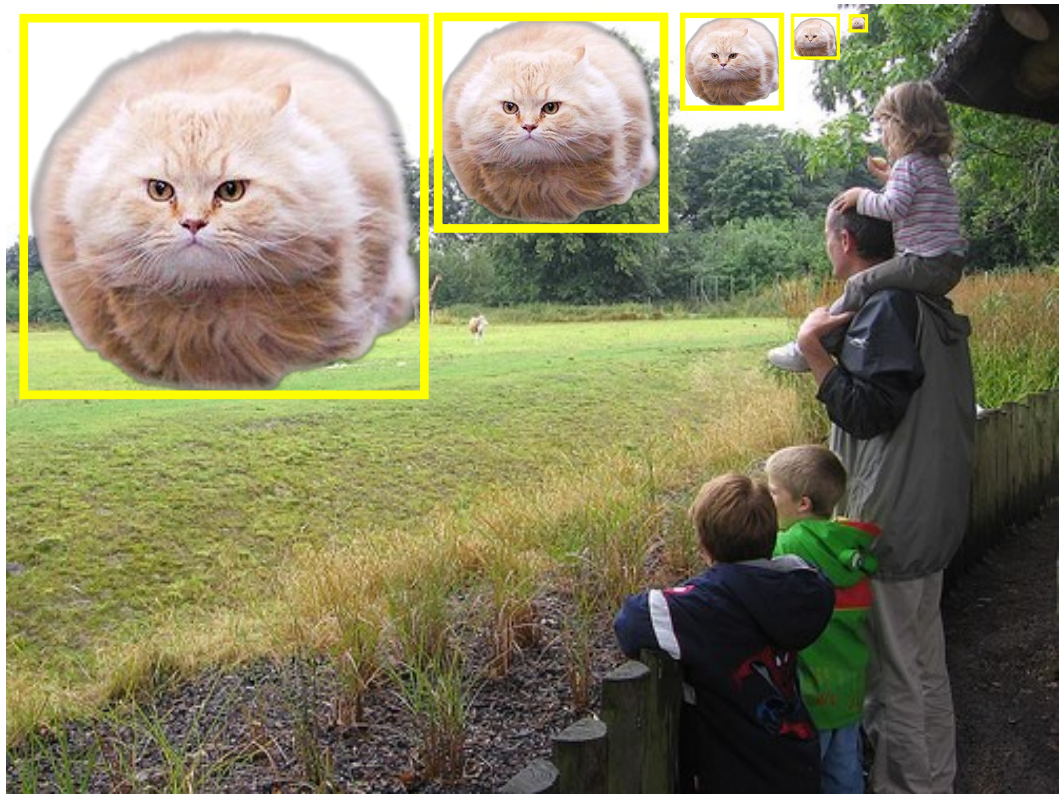


(c) Pyramidal feature hierarchy

Use the internal pyramid – *fast, suboptimal*

(E.g., ≈ SSD, …)

Strategy 4: Feature Pyramid Network

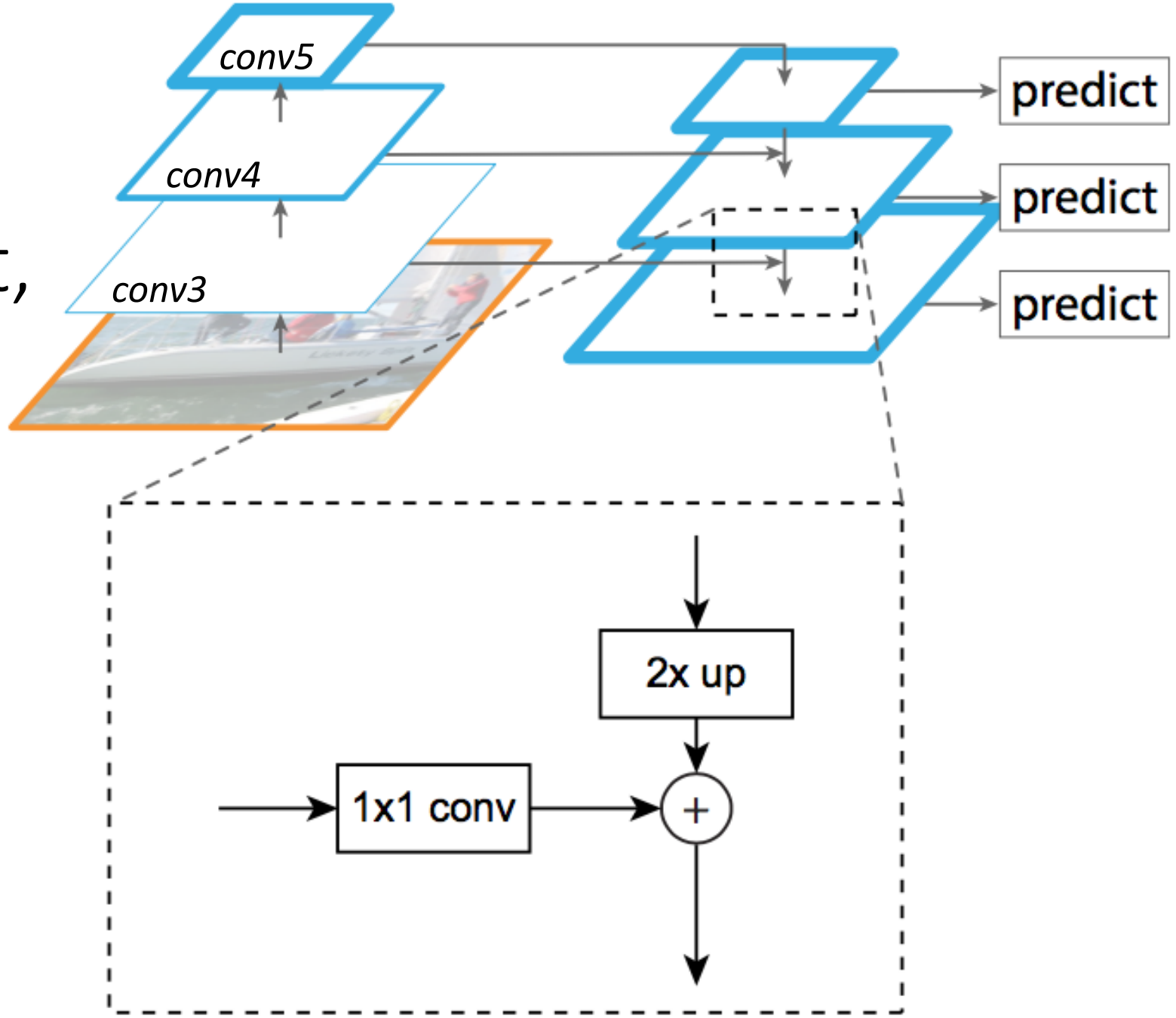


(d) Feature Pyramid Network

Top-down enrichment of high-res features – *fast, less suboptimal*

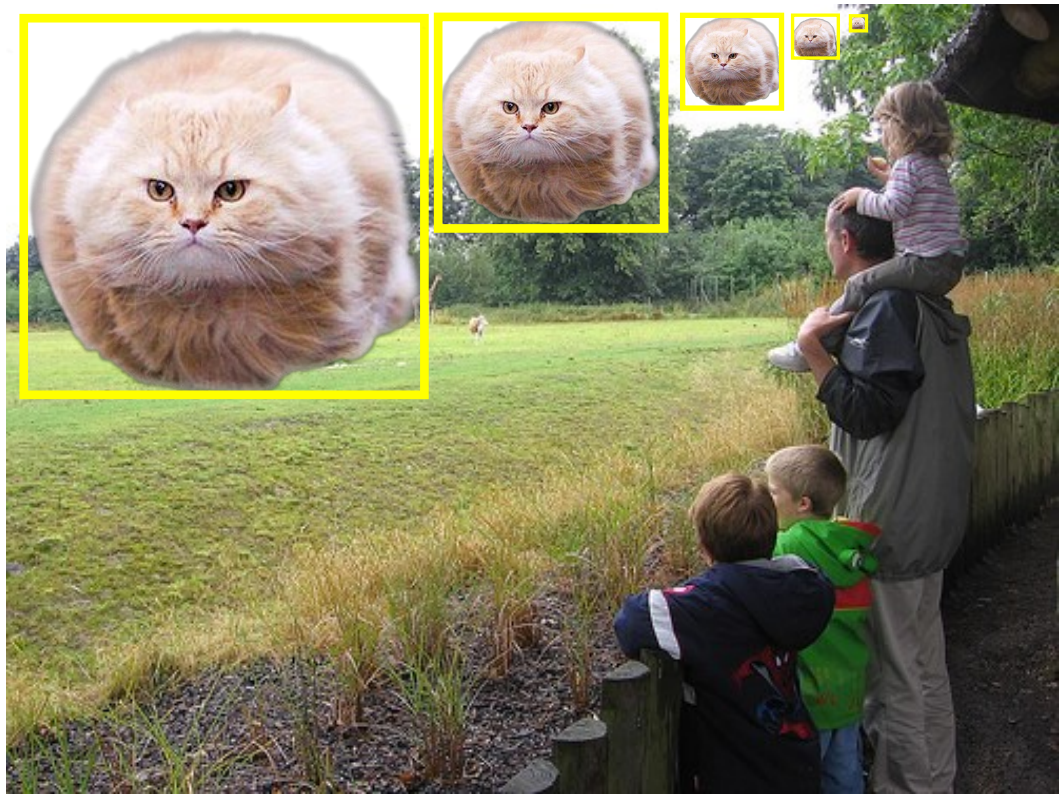
Lin et al. Feature Pyramid Networks for Object Detection. CVPR 2017.

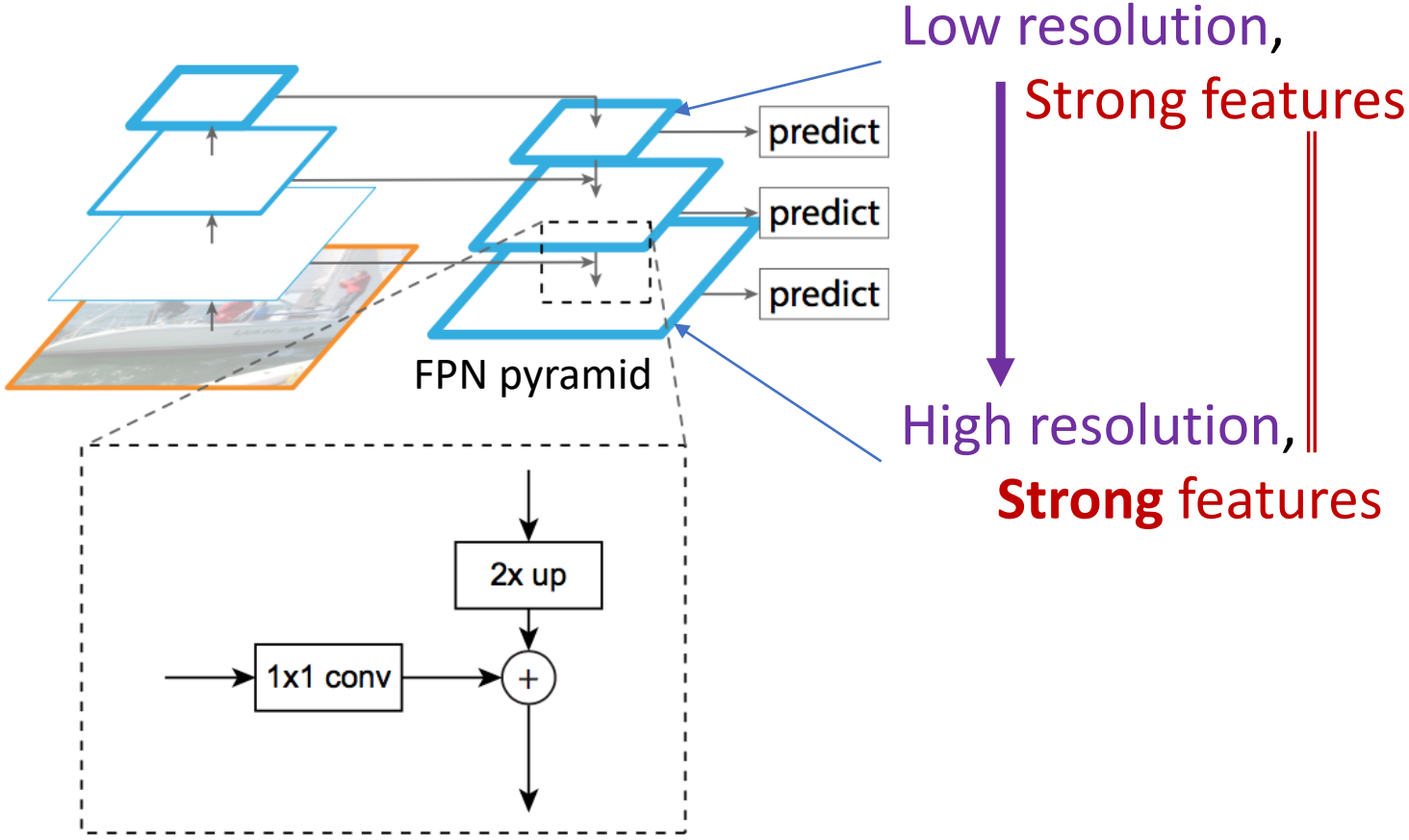
FPN: Light-weight, Top-down Refinement Module



Lin et al. Feature Pyramid Networks for Object Detection. CVPR 2017.

No Compromise on Feature Quality, Still Fast





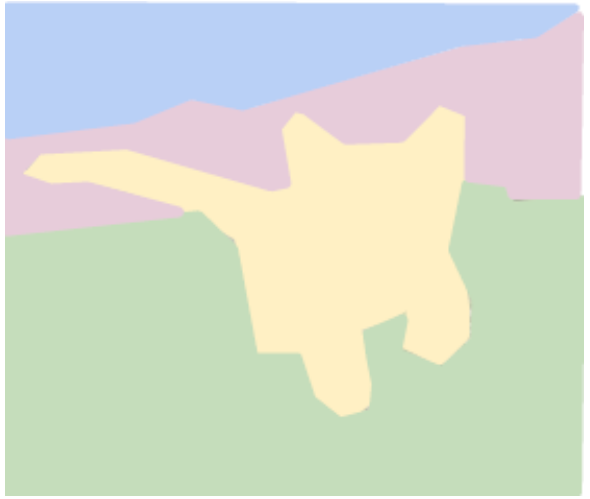
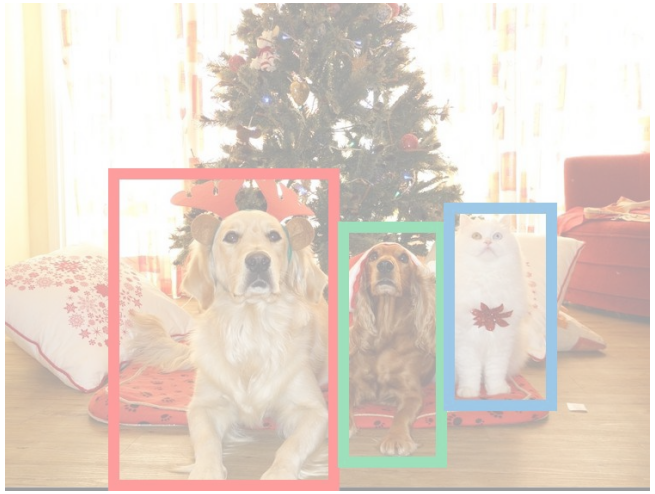
Lin et al. Feature Pyramid Networks for Object Detection. CVPR 2017. See also: Shrivastava's TDM.

FPN – A Generic Backbone Modification

Generates a feature pyramid—*useful in many applications!*

- RPN
- Fast/er R-CNN
- Mask R-CNN
- RetinaNet
- …

Instance Segmentation

Classification	Semantic Segmentation	Object Detection	**Instance Segmentation**
			
CAT	GRASS, CAT TREE, SKY	DOG, DOG, CAT	**DOG, DOG, CAT**
No spatial extent	No objects, just pixels	Multiple Object	

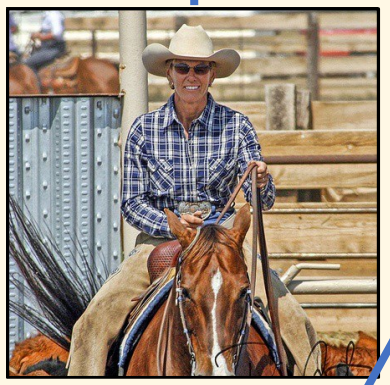
Faster R-CNN with a Feature Pyramid Network

Per-image computation

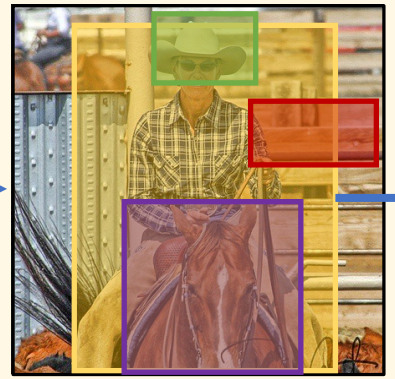
Per-region computation for each $r_i \in r(I)$

f_I = FPN(I)

I:



RPN(f_I)



RoIPool

MLP

Softmax clf.

Box regressor

The whole-image feature representation can be improved by making it *multi-scale*

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

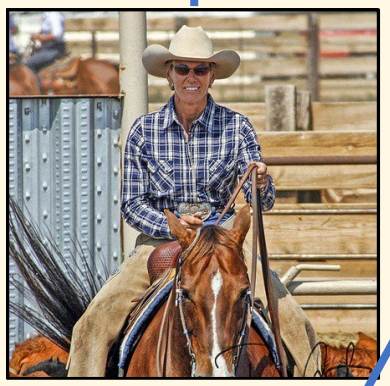
Mask R-CNN

Per-image computation

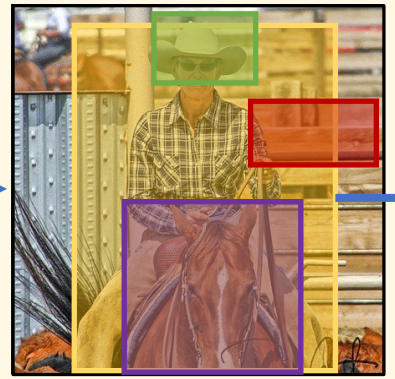
Per-region computation for each $r_i \in r(I)$

f_I = FPN(I)

I:



RPN(f_I)



RoIAlign

MLP

Softmax clf.

Box regressor

Cascaded heads (inference only)

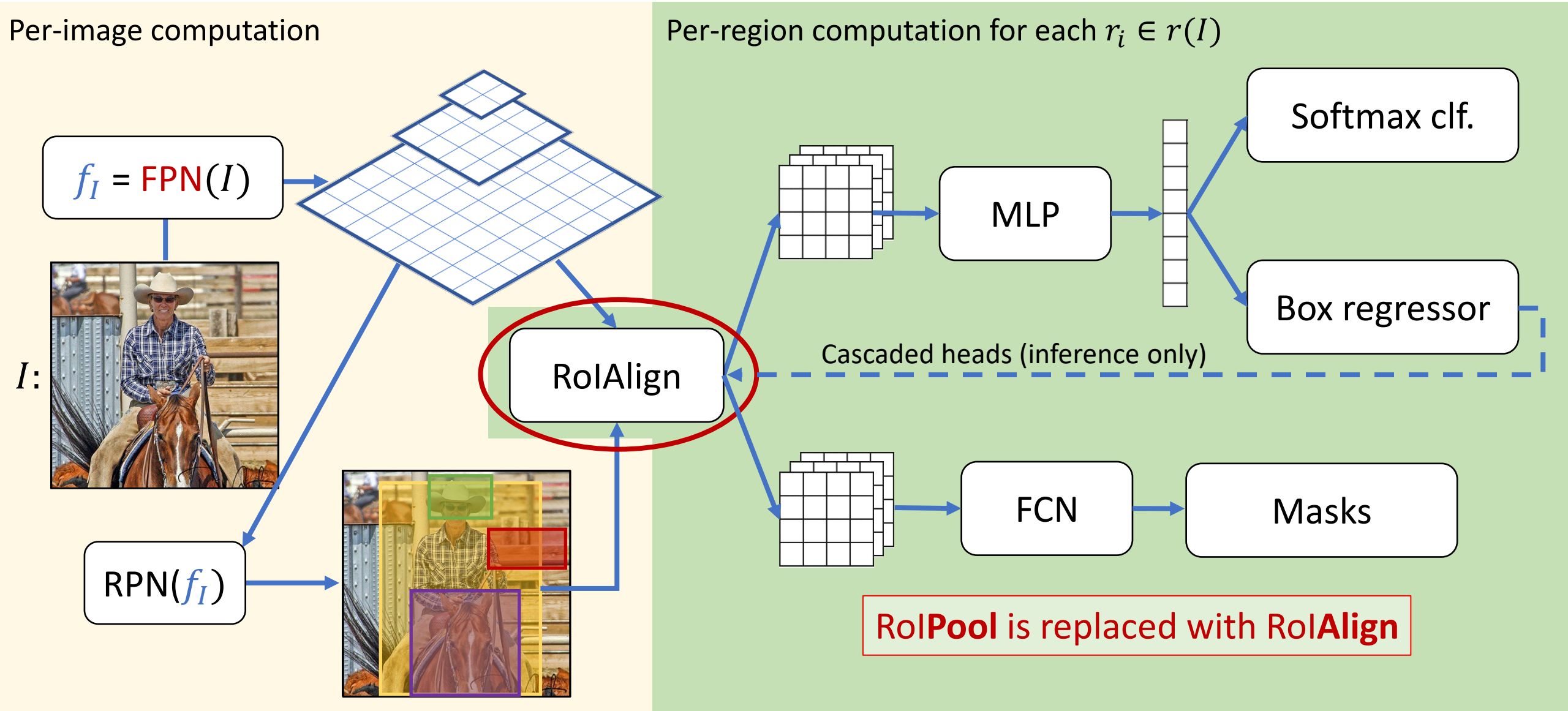
FCN

Masks

An additional head is added to predict instance-level segmentation masks

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

Mask R-CNN



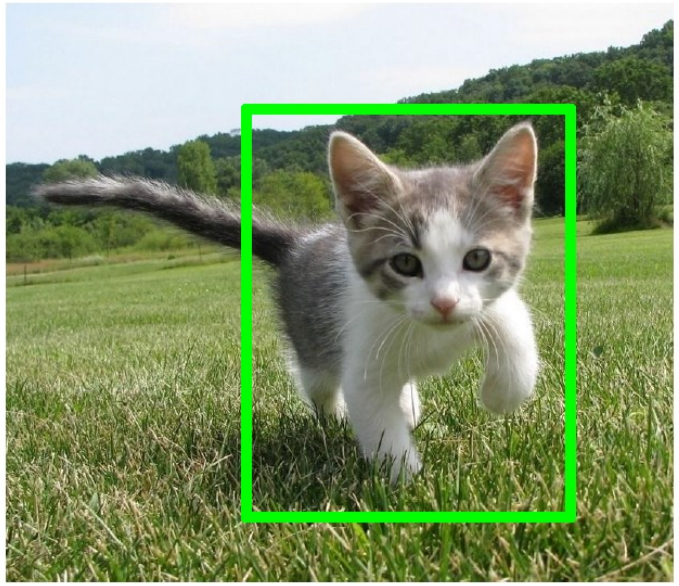
Girshick. Fast R-CNN. ICCV 2015.

Cropping Features: RoIPool

Project proposal onto features

"Snap" to grid cells

Divide into 2x2 grid of (roughly) equal subregions



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

CNN

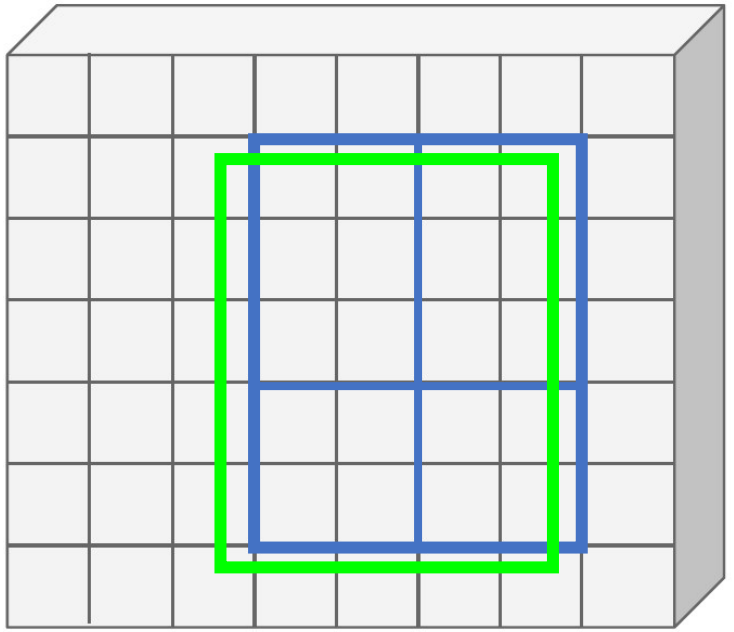
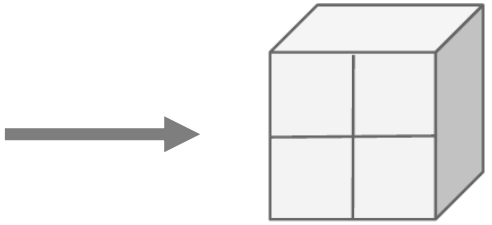


Image features: C x H x W
(e.g. 512 x 20 x 15)

Max-pool within each subregion



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

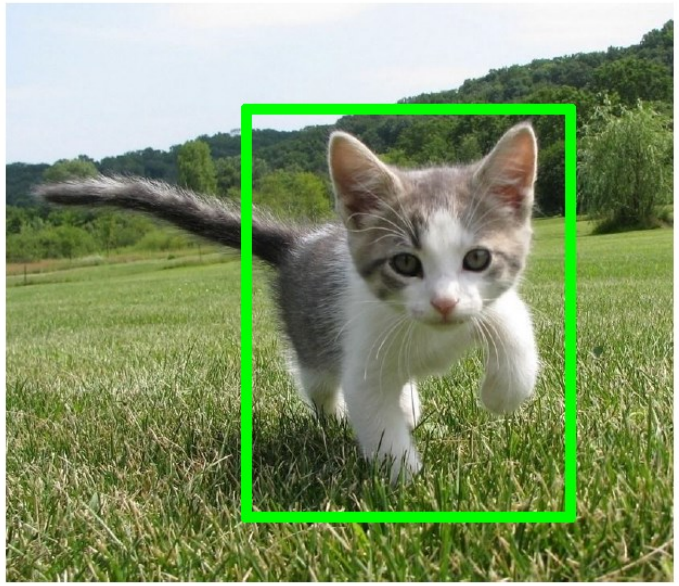
Problem: Region features slightly misaligned

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

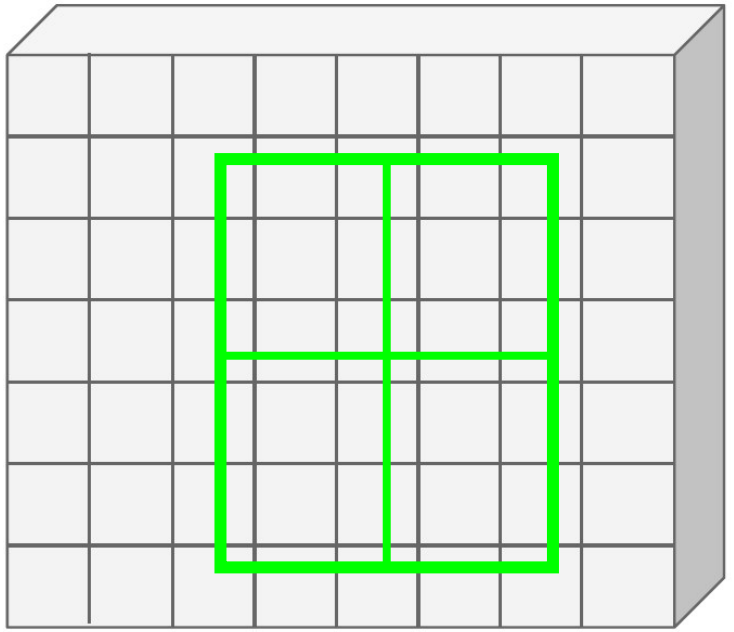
Cropping Features: RoIAlign

Project proposal onto features

No “snapping”!



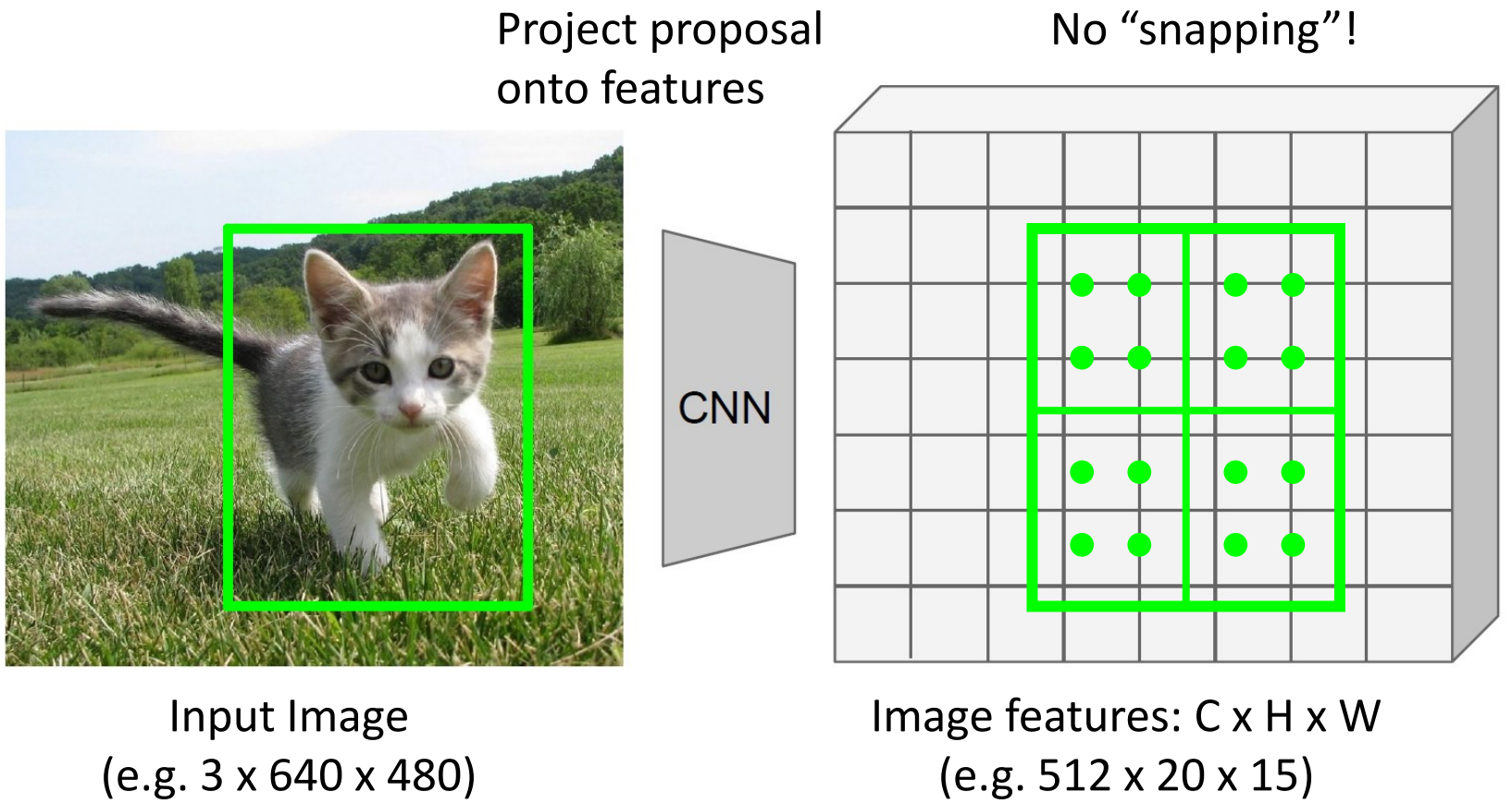
CNN



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

Image features: C x H x W
(e.g. 512 x 20 x 15)

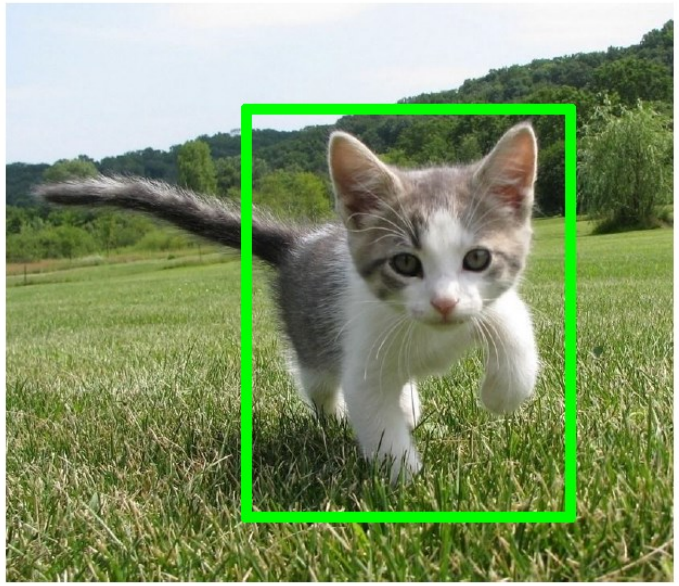
Cropping Features: RoIAlign



Cropping Features: RoIAlign

Project proposal onto features

No “snapping”!



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

CNN

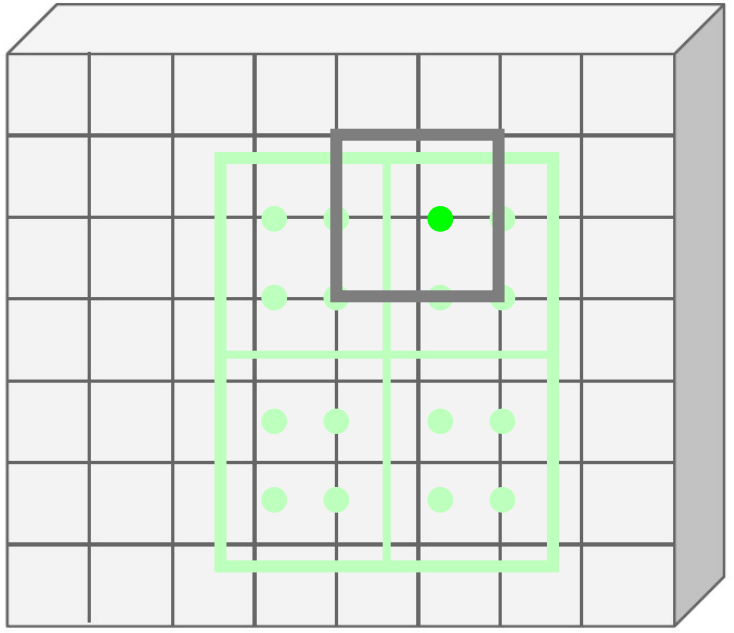
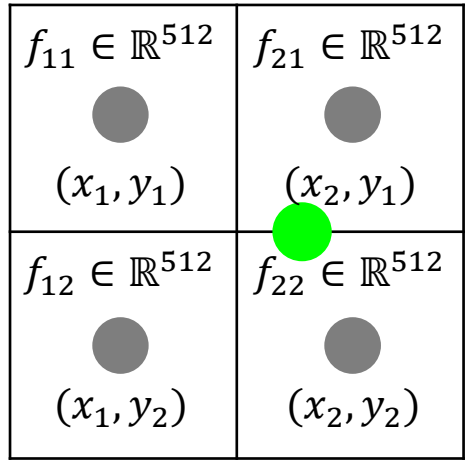


Image features: C x H x W
(e.g. 512 x 20 x 15)



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

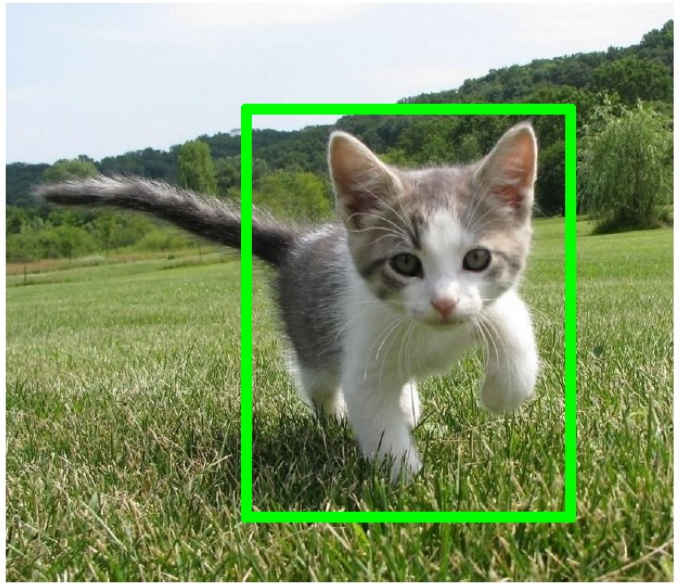
$$f_{xy} = \sum_{i,j=1}^{2} f_{i,j} \max(0, 1 - |x - x_i|) \max(0, 1 - |y - y_j|)$$

He et al, “Mask R-CNN”, ICCV 2017

Cropping Features: RoIAlign

Project proposal onto features

No “snapping”!



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

CNN

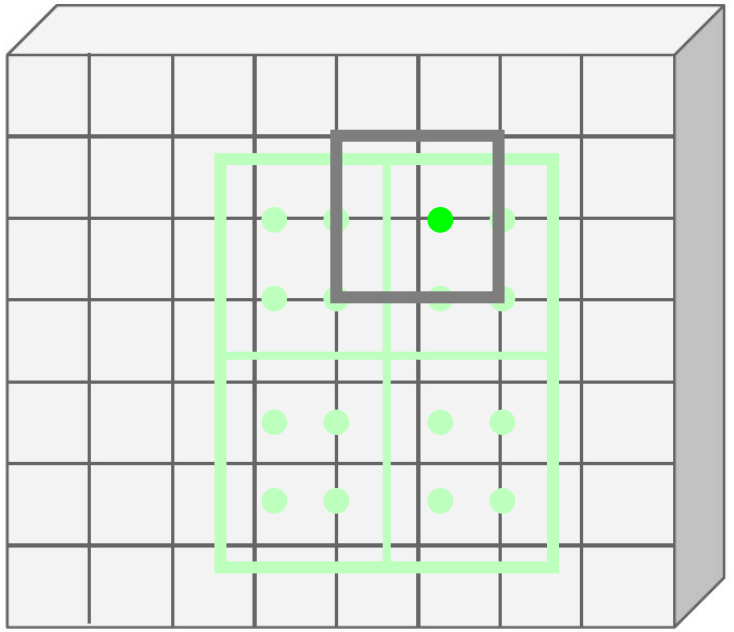
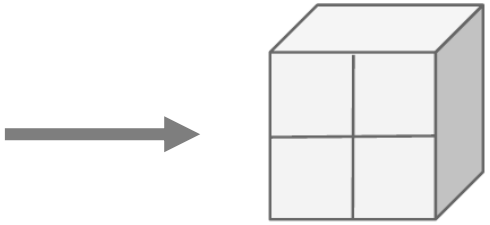


Image features: C x H x W
(e.g. 512 x 20 x 15)

Max-pool within each subregion



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.

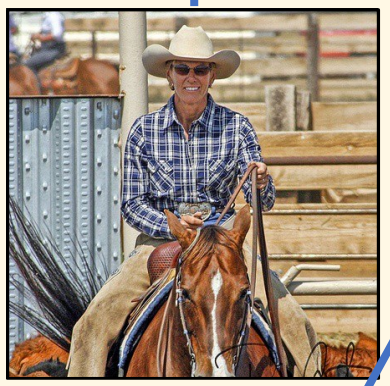
Mask R-CNN

Per-image computation

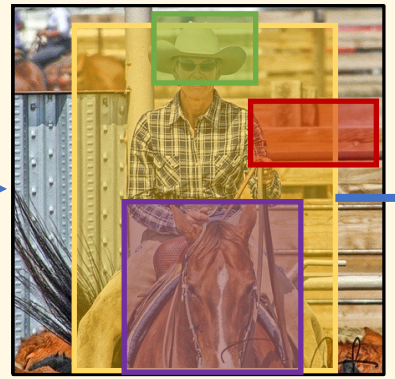
Per-region computation for each $r_i \in r(I)$

f_I = FPN(I)

I:



RPN(f_I)



RoIAlign

MLP

Softmax clf.

Box regressor

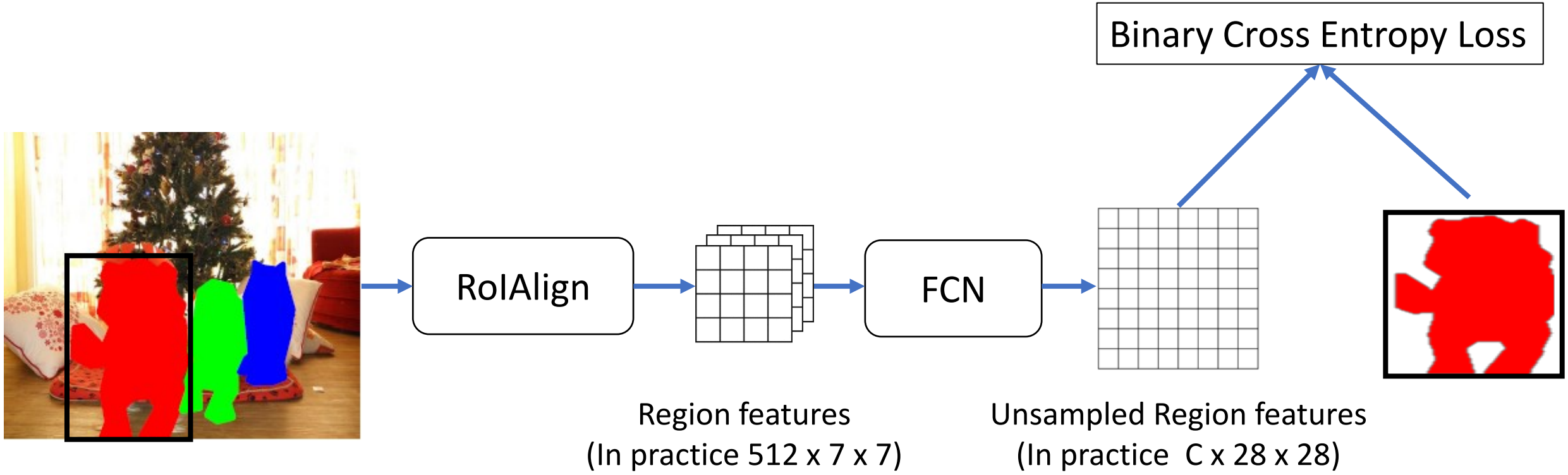
Cascaded heads (inference only)

FCN

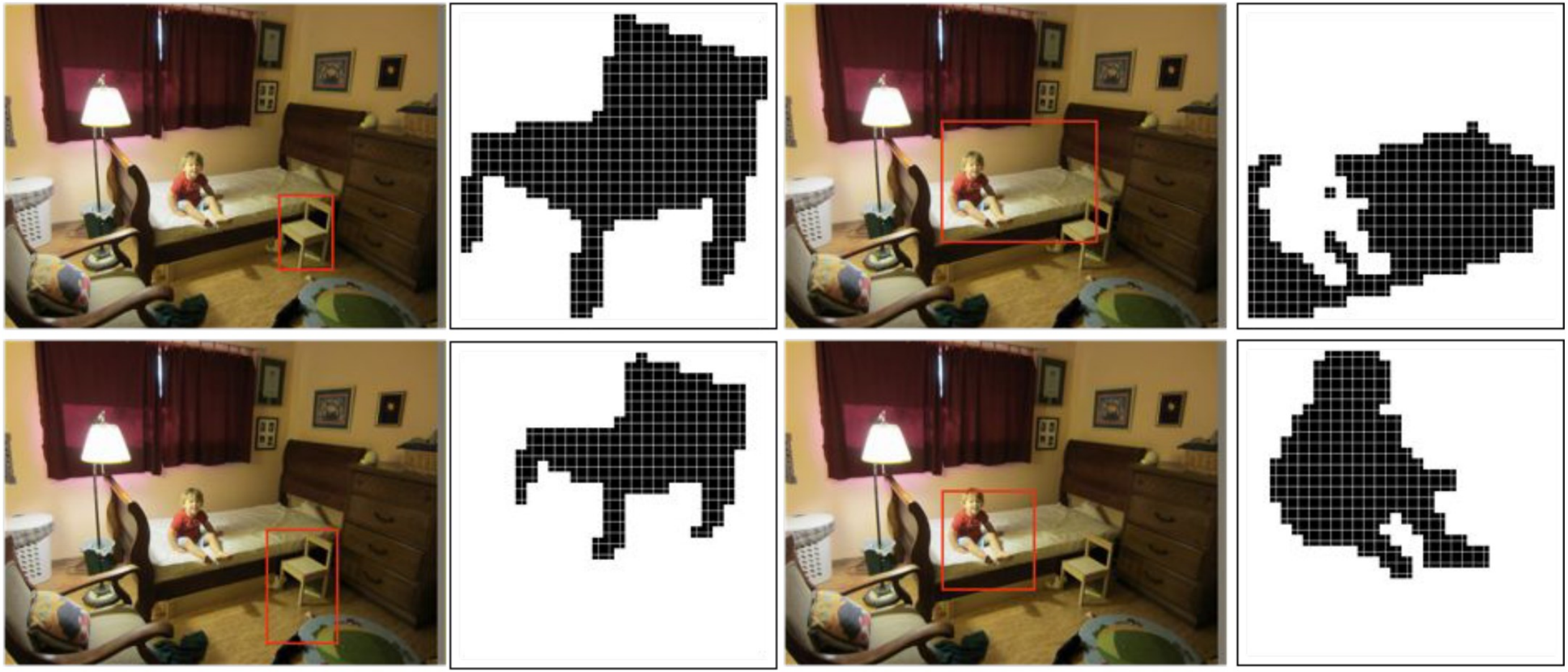
Masks

Training the mask head

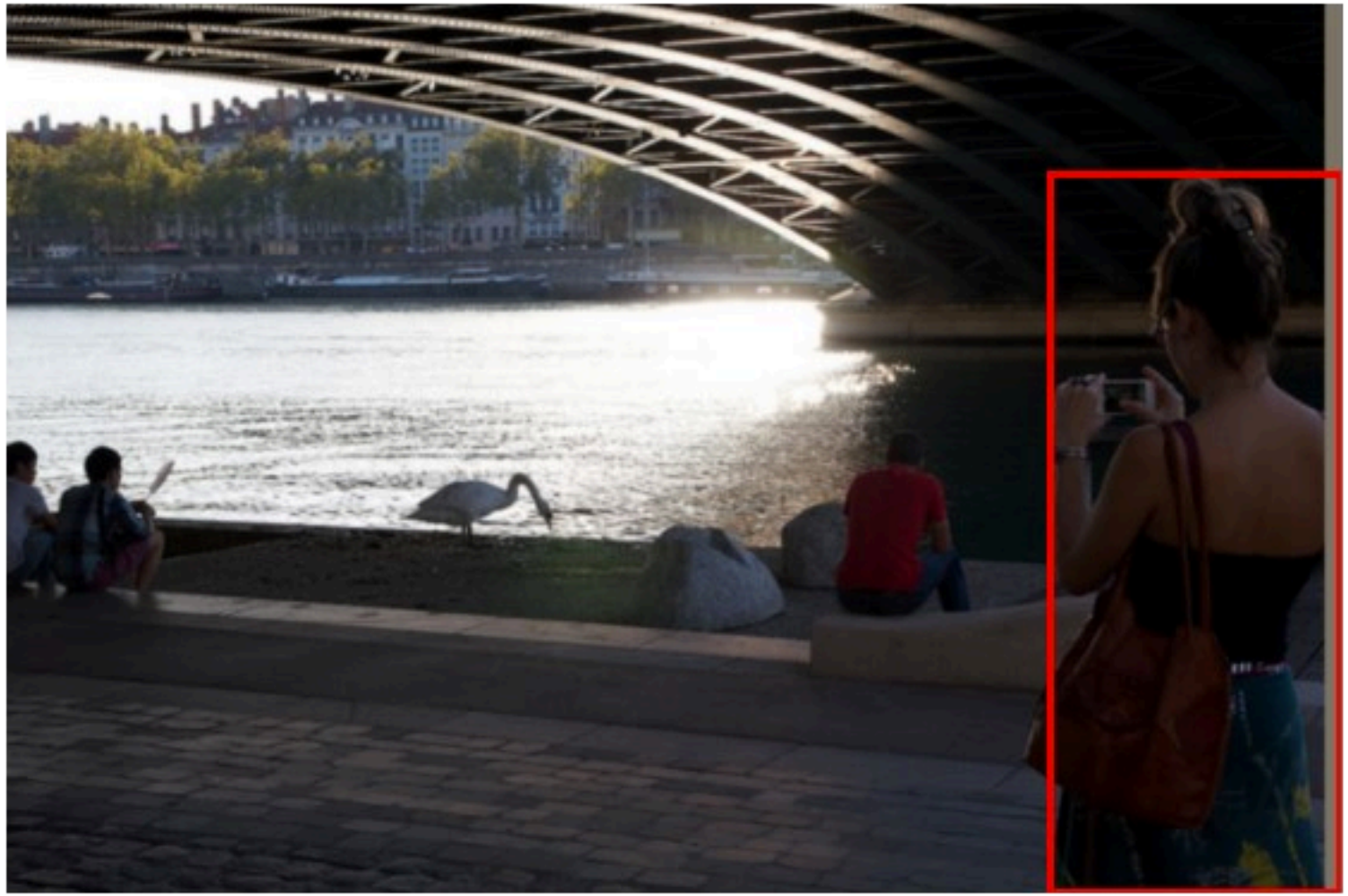
Training the mask head



Mask R-CNN: Example Mask Training Targets

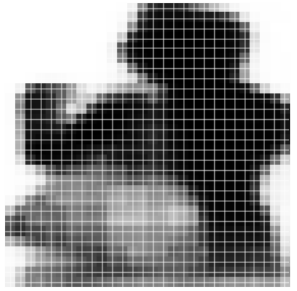


Mask Prediction



Validation image with box detection shown in red

28x28 soft prediction

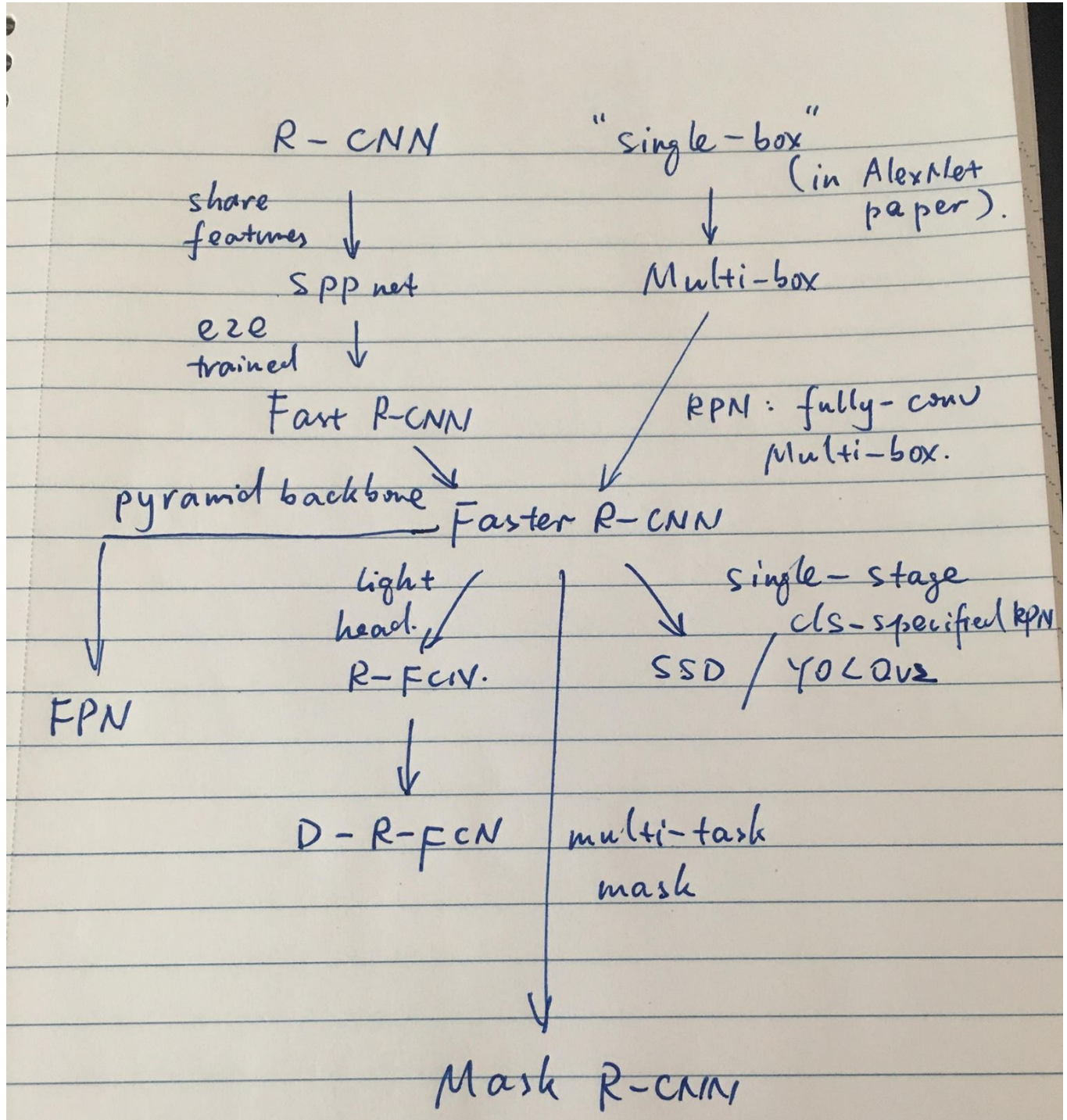


Resized soft prediction

Final mask


person1.00
person1.00
person1.00
person.98
surfboard1.00
person1.00
person.91
surfboard1.00
surfboard.98
surfboard1.00
person.74
surfboard1.00


person.88
person1.00
tv.98
tv.84
person1.00
person1.00
bottle.97
wine glass.99
dining table.95
wine glass1.00
wine glass1.00


R-CNN
"single-box"
(in AlexNet paper).
share features
SPP net
Multi-box
e2e trained
Fast R-CNN
RPN: fully-conv Multi-box.
pyramid backbone
Faster R-CNN
light head.
single-stage
cls-specified RPN
FPN
R-FCN.
SSD / YOLOv2
D-R-FCN
multi-task mask
Mask R-CNN