MAI4CAREU Master programmes in Artificial Intelligence 4 Careers in Europe

University of Cyprus

University of Cyprus

MAI645 - Machine Learning for Graphics and Computer Vision

Andreas Aristidou, PhD

Spring Semester 2023

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Image Classification: *CNN Architectures*

These notes are based on the work of Fei-Fei Li, Jiajun Wu, Ruohan Gao,

CS231 - Deep Learning for Computer Vision

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Deep neural networks for image classification

Deep learning has proven to take computer vision tasks to an even higher level of accuracy and efficiency, all thanks to convolutional neural networks (CNNs). The aim of it is to emulate the neural networks of the human mind in order to complete specific computer processes with minimal human interference. The variety of layers, starting with the input layer, to the hidden inner layers, and output layer are what make the network considered "deep." In brief, this is how image classification is done via CNNs:

- The input image is fed into the network.
- Various filters are applied to the image in order to generate a feature map.
- A pooling layer is applied to each of those maps.
- The pooled layers are flattened into a vector, then that vector is connected to the neural network.
- The final fully-connected output layer with the classified features is received.

Fully grasping the use of CNNs for image classification requires a much deeper dive into the technical aspect of the model. That deserves a separate crash course of its own if you aim to learn beyond the basics of image classification.

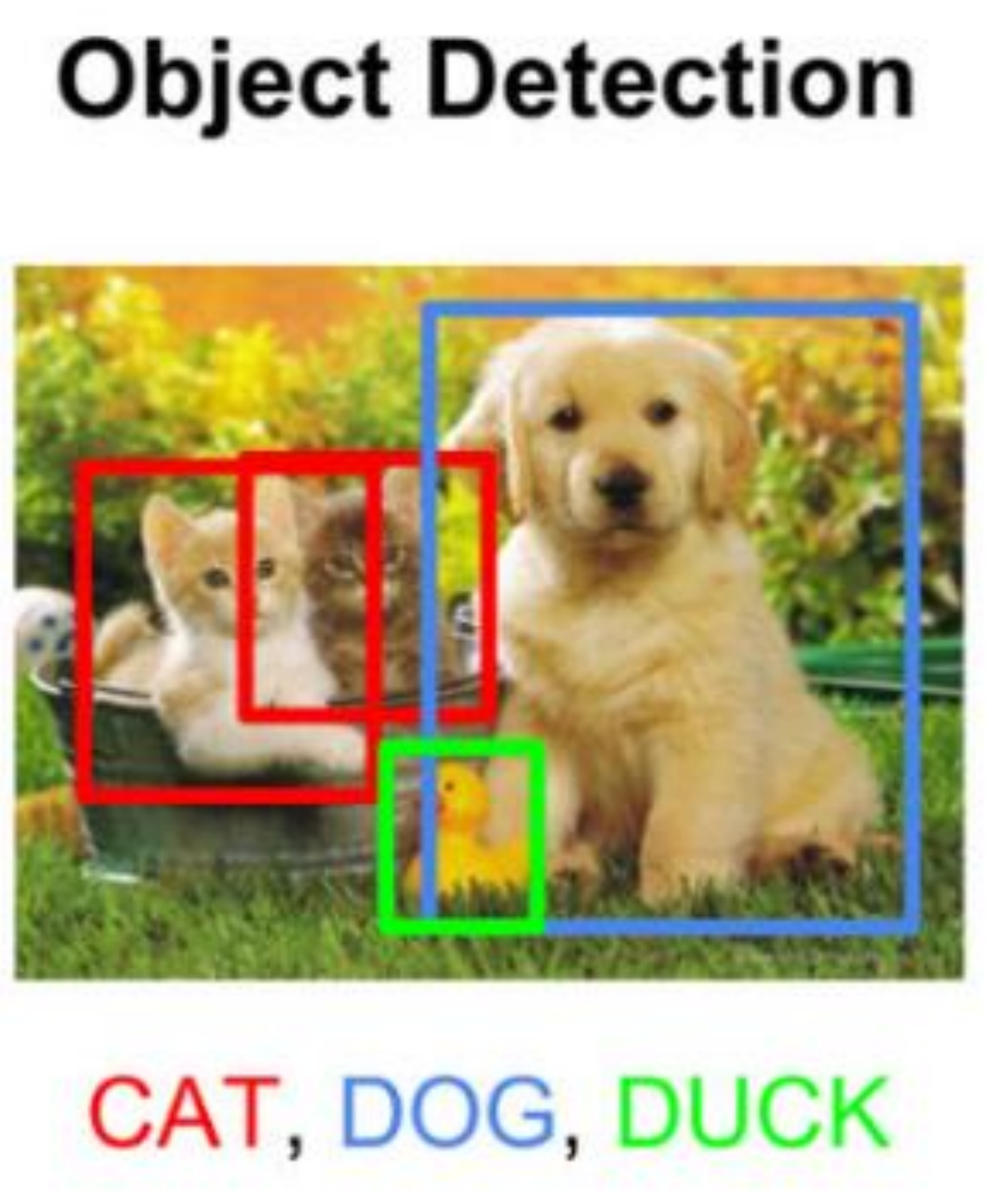
This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Object detection

Object detection is a fundamental task in computer vision where the goal is to locate and classify objects within an image or video.

It is a more advanced form of image classification, where instead of just identifying the class of an entire image, they identify multiple instances of multiple classes within an image and locate them with a bounding box; in other words, it deals with more realistic cases in which multiple objects may exist in an image

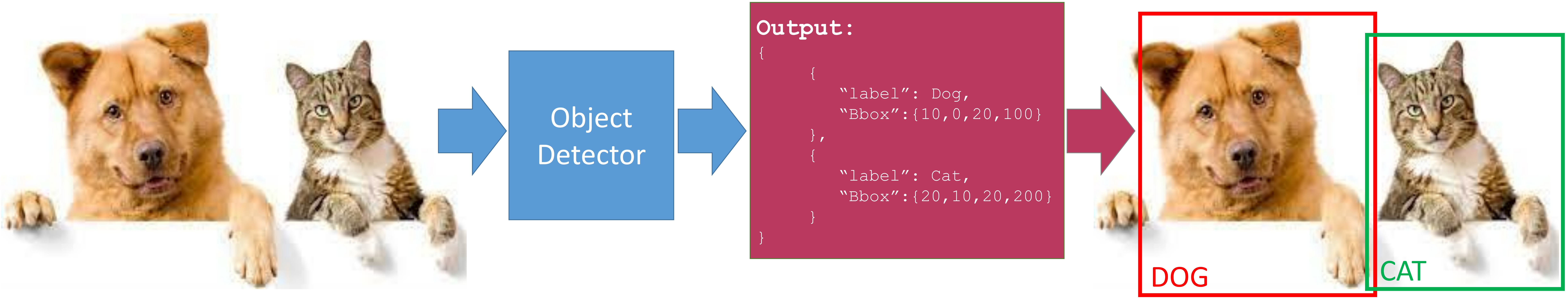
Object detection algorithms can be used in a variety of applications, such as self-driving cars, security systems, and augmented reality.



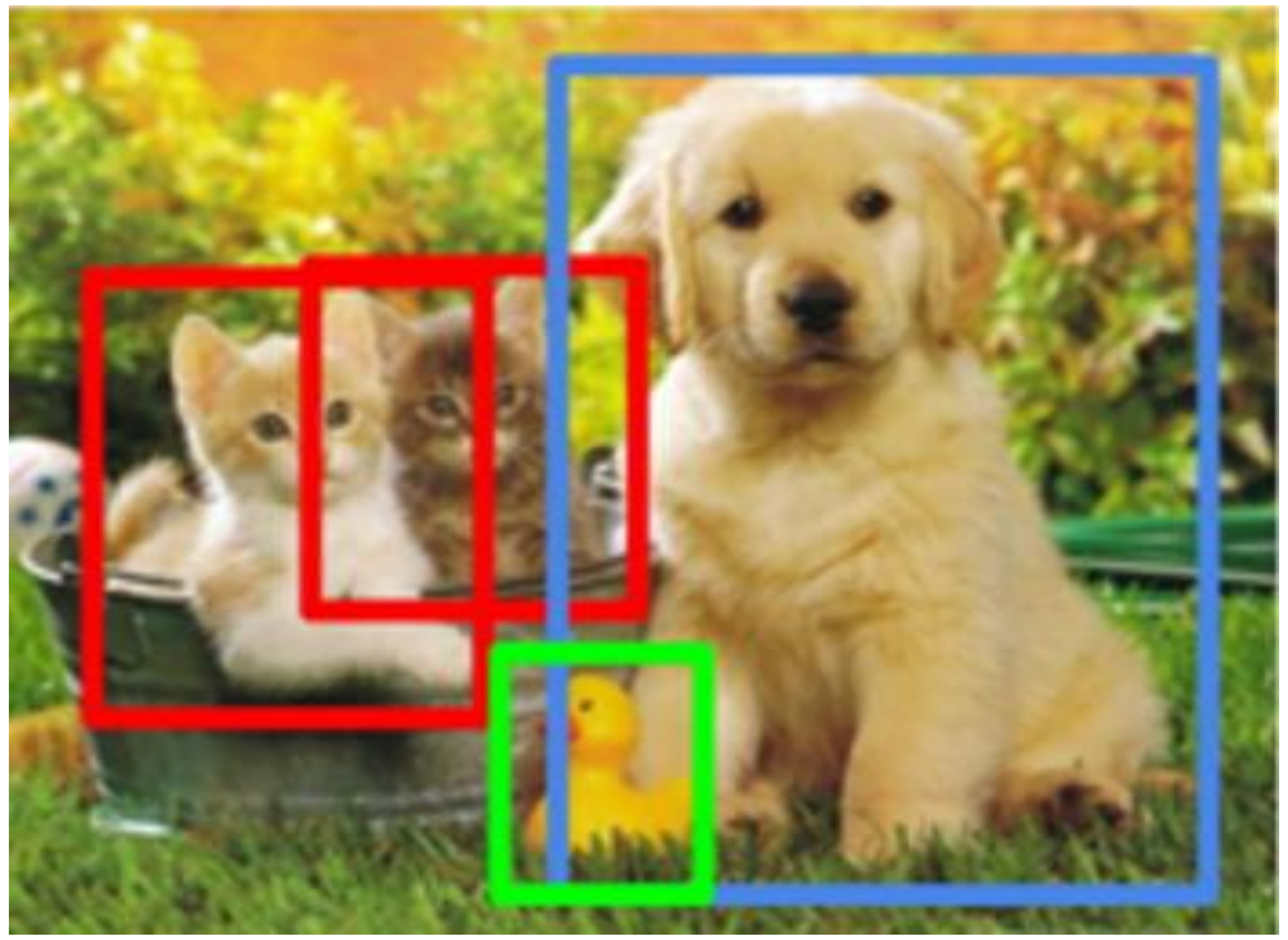
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Object detection



Differences

Image Classification	Localization	Object Detection
		
CAT	CAT	CAT, DUCK, DOG

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

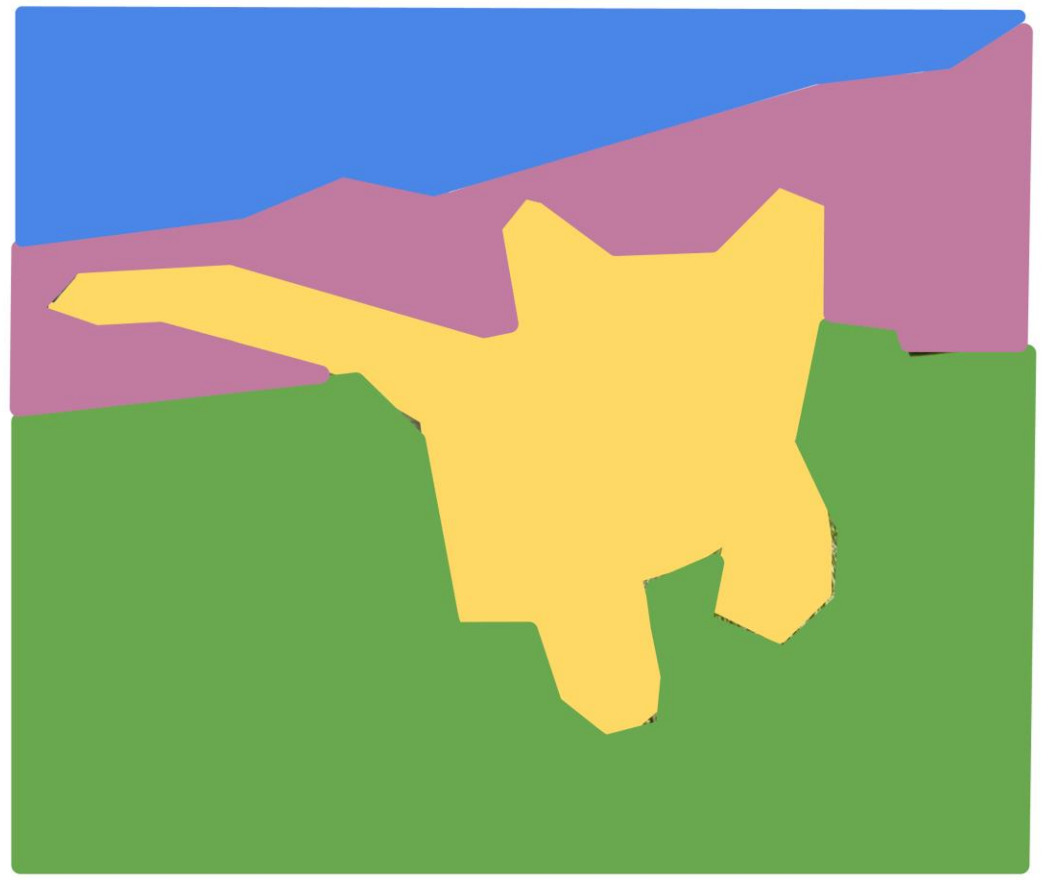
Computer Vision Tasks

Classification

CAT

No spatial extent

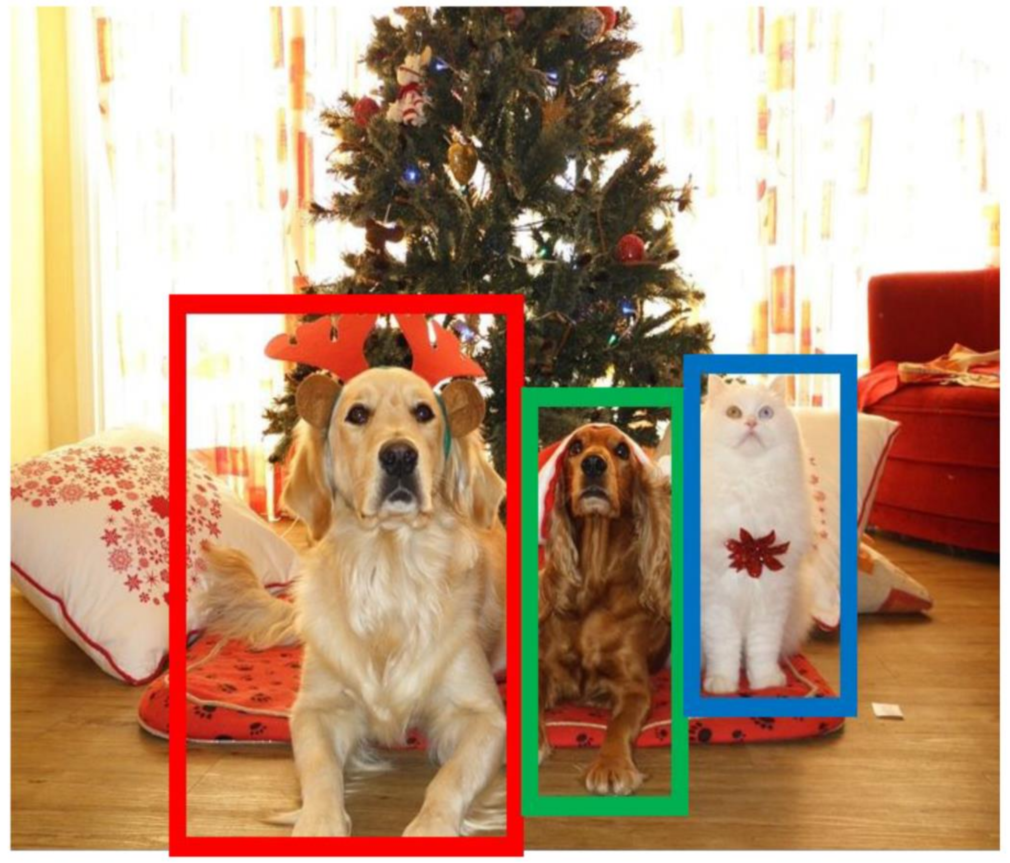
Semantic Segmentation



GRASS, CAT, TREE, SKY

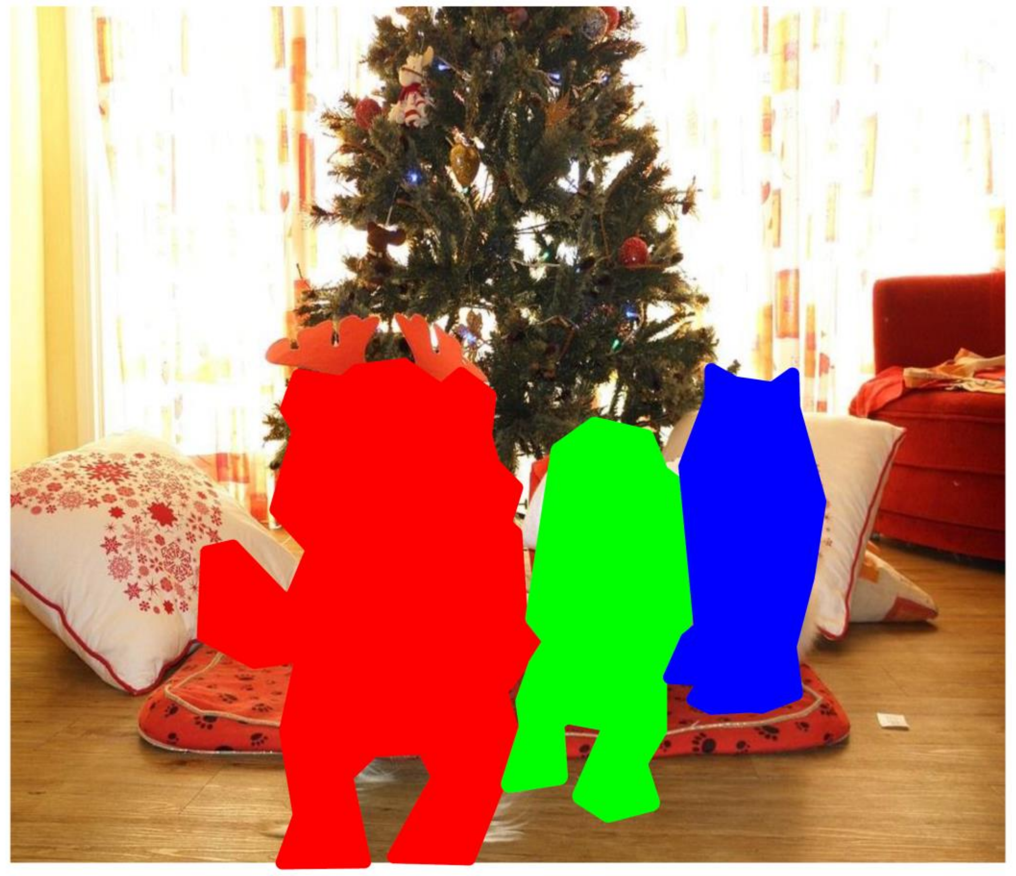
No objects, just pixels

Object Detection



DOG, DOG, CAT

Instance Segmentation

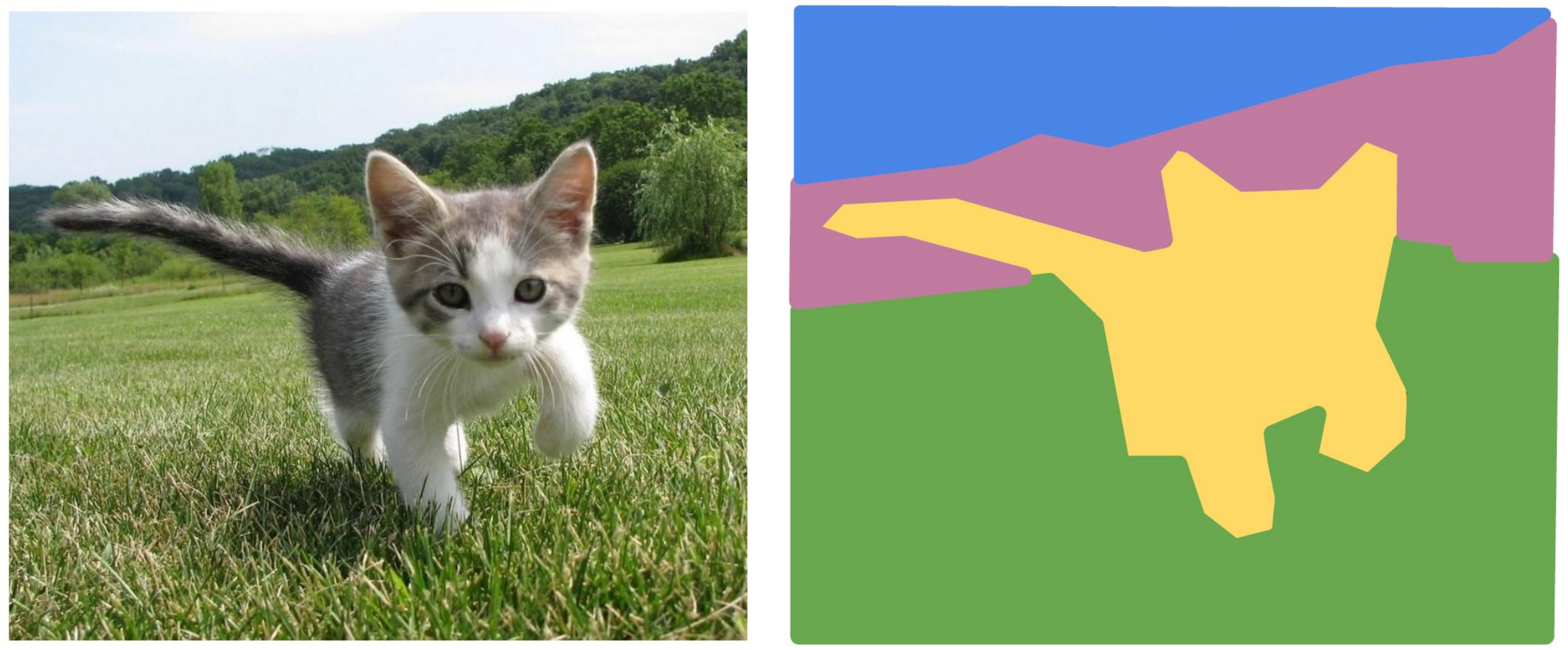


DOG, DOG, CAT

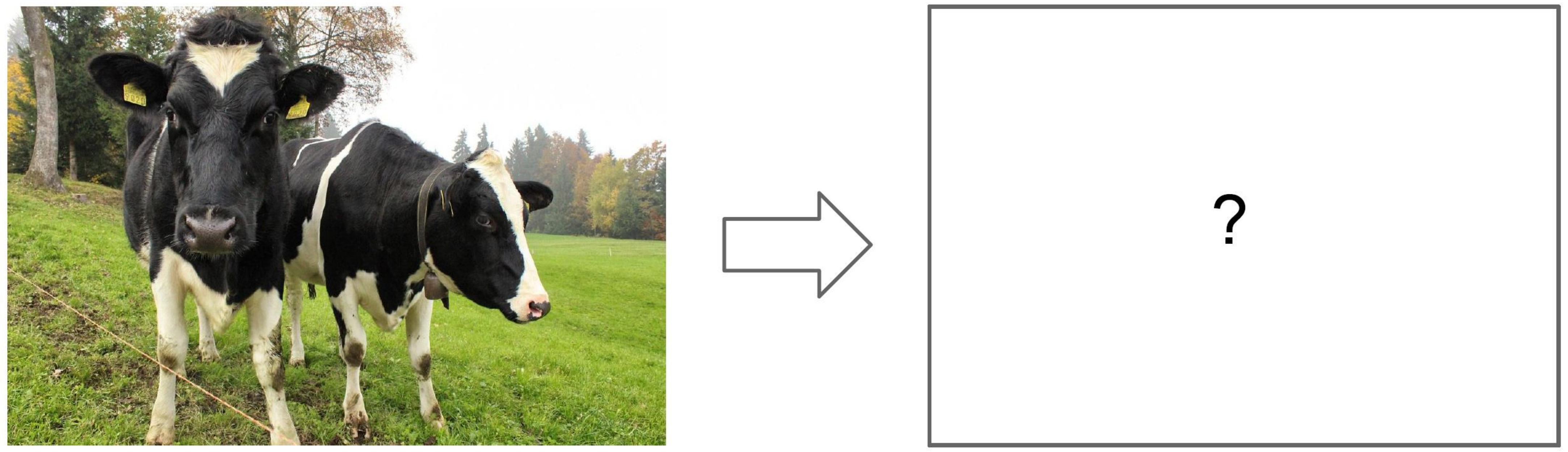
Multiple Object

This image is CC0 public domain

Semantic Segmentation: *The Problem*



GRASS, CAT, TREE, SKY, ...



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

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

Semantic Segmentation: *The Problem*

Semantic segmentation is the process of dividing an image into multiple segments and assigning each segment a label or a class. The sliding window is an idea for semantic segmentation that involves breaking an image into smaller sections, or windows, and processing each window separately to identify the objects and their boundaries.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Semantic Segmentation: *Sliding window*

Full image

?

Impossible to classify without context

Q: how do we include context?

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Semantic Segmentation: *Sliding window*

In the sliding window approach, a window of a fixed size moves across the image, classifying each portion of the image that it covers. The classification is performed using a machine learning model that has been trained on a dataset of annotated images, where each pixel in the image is labeled with the corresponding object class.

As the window slides across the image, it classifies each portion of the image that it covers, producing a segmentation map that identifies the object classes and their boundaries within the image. The resulting segmentation map can be used for a variety of tasks, such as object recognition, image segmentation, and scene understanding.

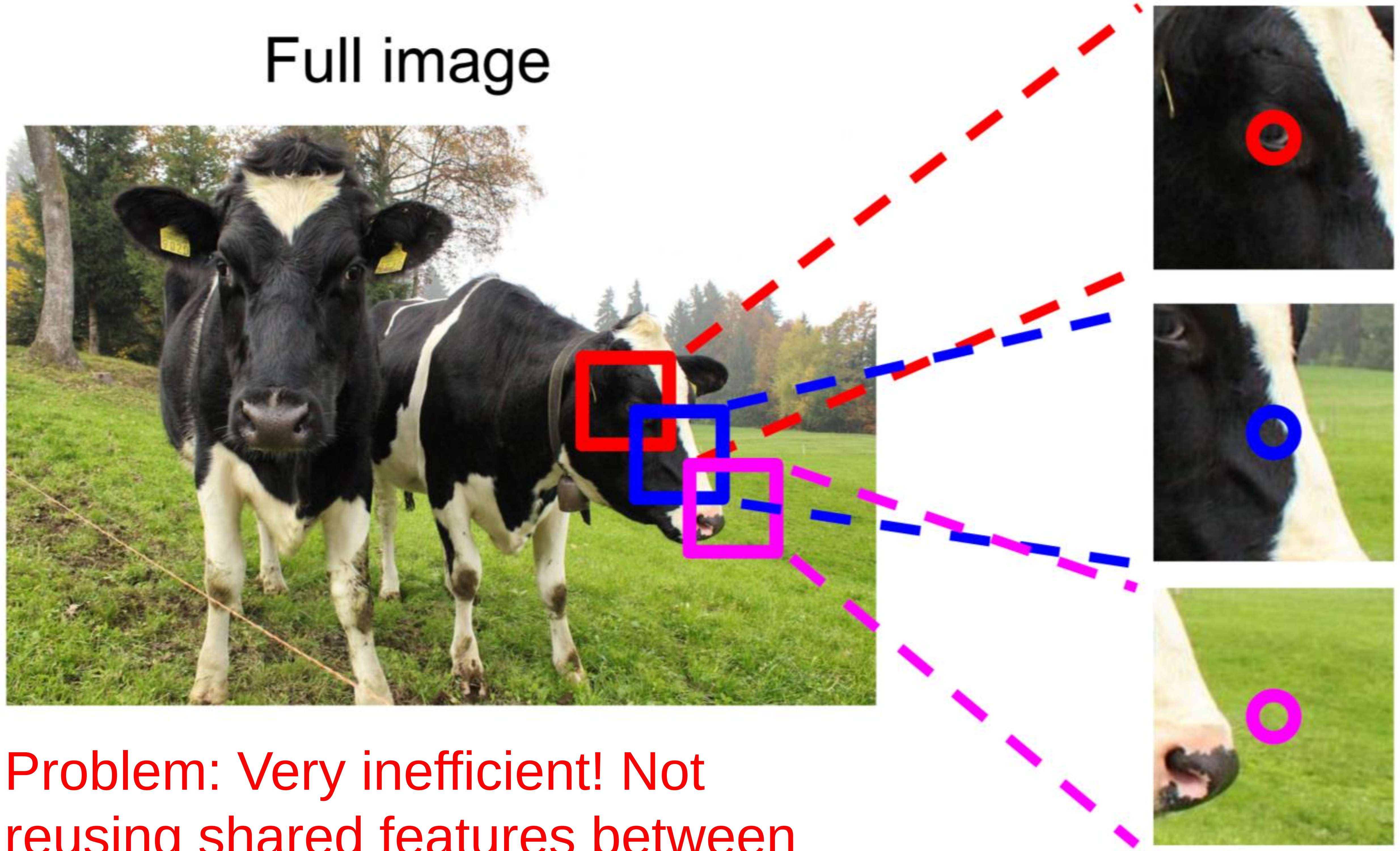
While the sliding window approach can be effective for semantic segmentation, it can also be computationally expensive, especially for larger images. To address this issue, researchers have developed more efficient algorithms, such as convolutional neural networks (CNNs), that can perform semantic segmentation with high accuracy and lower computational cost.

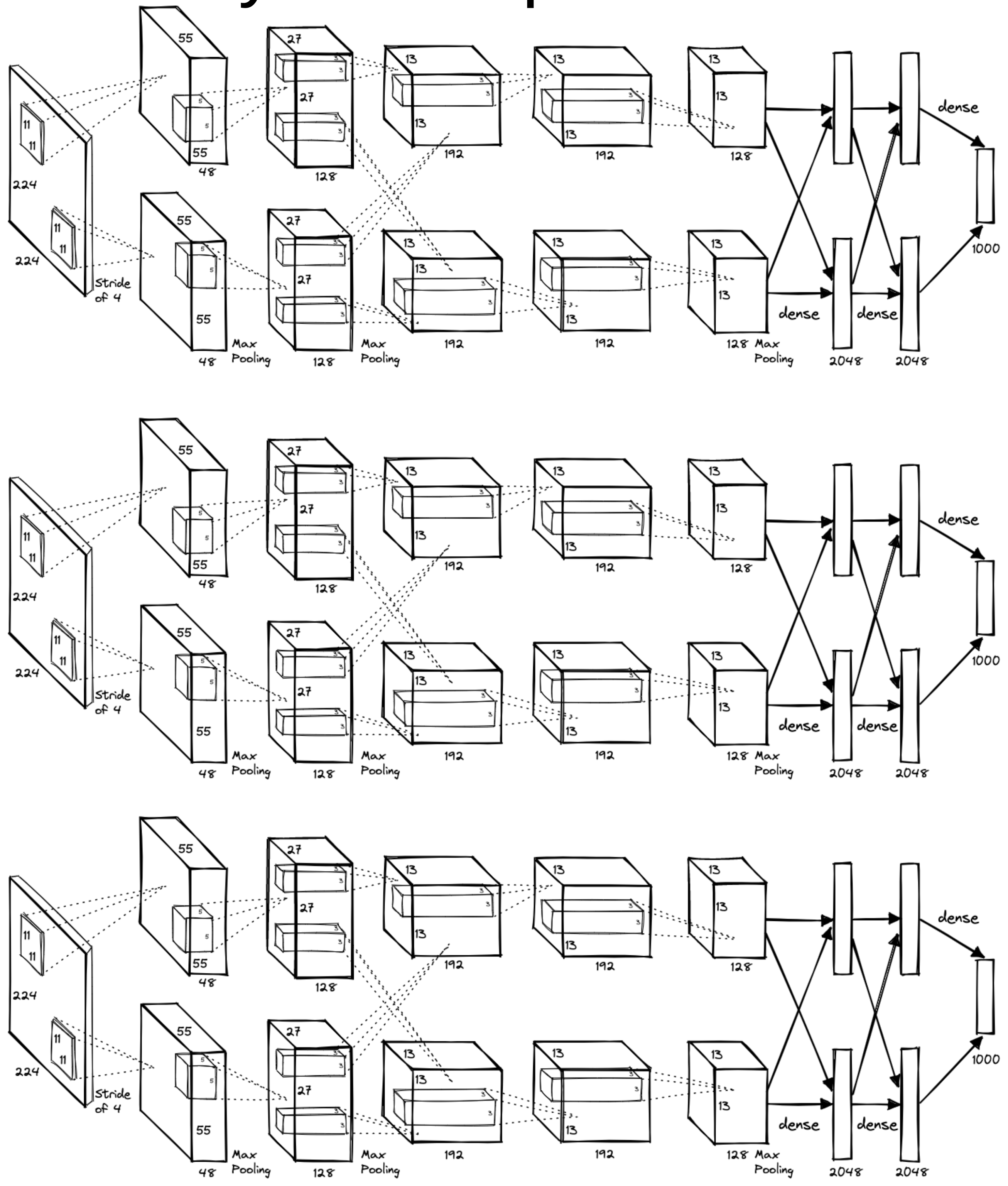
Semantic Segmentation: *Sliding window*

Full image

Extract patch

Classify center pixel with CNN





Cow

Cow

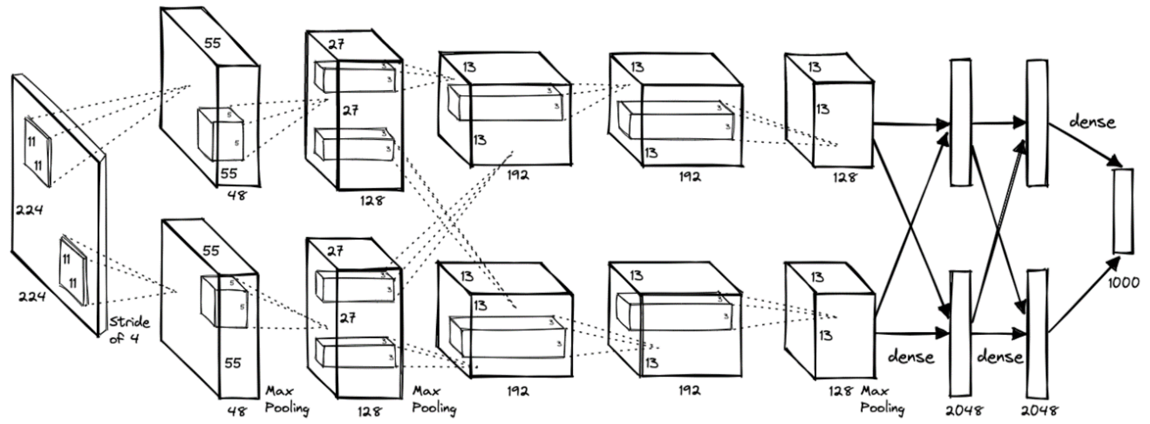
Grass

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

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

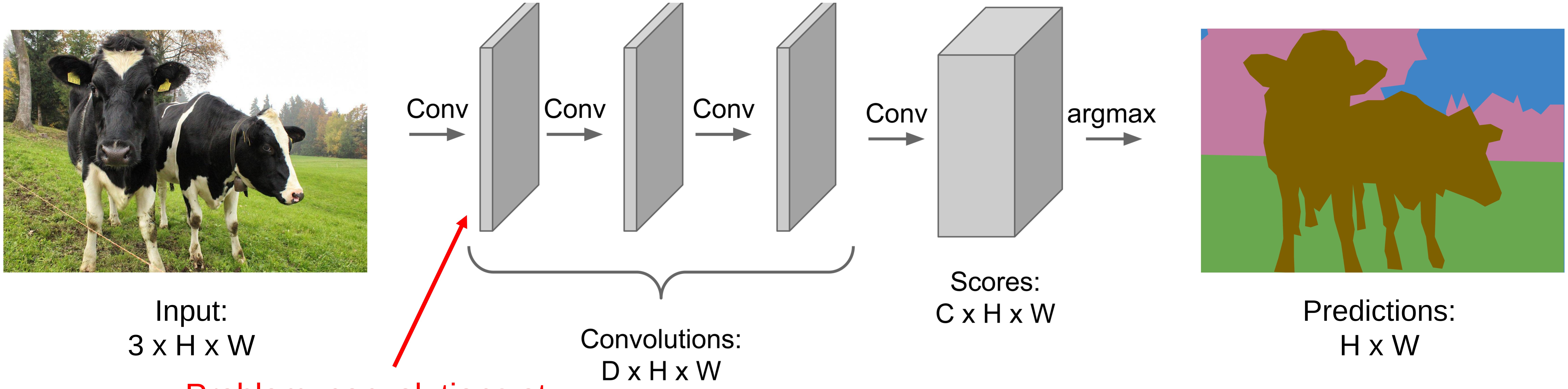
Semantic Segmentation Idea: *Convolution*



An intuitive idea: encode the entire image with conv net, and do semantic segmentation on top.

Problem: classification architectures often reduce feature spatial sizes to go deeper, but semantic segmentation requires the output size to be the same as input size.

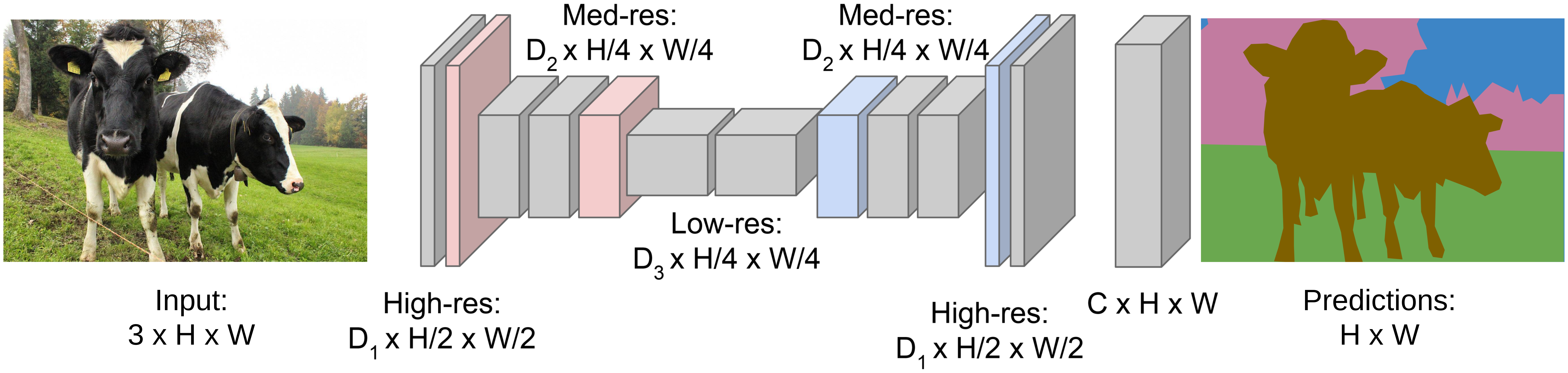
Semantic Segmentation Idea: *Convolution*



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

Design a network with only convolutional layers without downsampling operators to make predictions for pixels all at once!

Semantic Segmentation Idea: *Fully Convolutional*



Downsampling:
Pooling, strided convolution

Upsampling:
???

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

Co-financed by the European Union
Connecting Europe Facility
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

In-Network upsampling: *"Unpooling"*

Unpooling is a technique used in convolutional neural networks (CNNs) to increase the resolution of the feature map while preserving the spatial information and avoiding the introduction of artifacts.

Pooling operations in CNNs reduce the spatial dimension of the feature map by down-sampling it, which makes it difficult to reconstruct the original image. Unpooling, on the other hand, performs the inverse operation of pooling, which means it restores the original spatial dimensions of the feature map by upsampling it. The unpooling operation is usually performed in conjunction with a convolutional layer, where each pixel in the output feature map is associated with a receptive field in the input feature map. During the unpooling operation, the output feature map is first resized to match the size of the input feature map, and then the value of each pixel in the output feature map is distributed over its associated receptive field in the input feature map.

There are several approaches to unpooling, such as **nearest-neighbor interpolation**, **bilinear interpolation**, and **max unpooling**. In-network upsampling with unpooling is a useful technique for semantic segmentation tasks because it can improve the resolution of the feature map, which allows for better localization of objects and more precise segmentation boundaries.

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

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

In-Network upsampling: *"Max Unpooling"*

Max Pooling
Remember which element was max!

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

Input: 4 x 4

5	6
7	8

Output: 2 x 2

→ … →

Rest of the network

Max Unpooling
Use positions from pooling layer

1	2
3	4

Input: 2 x 2

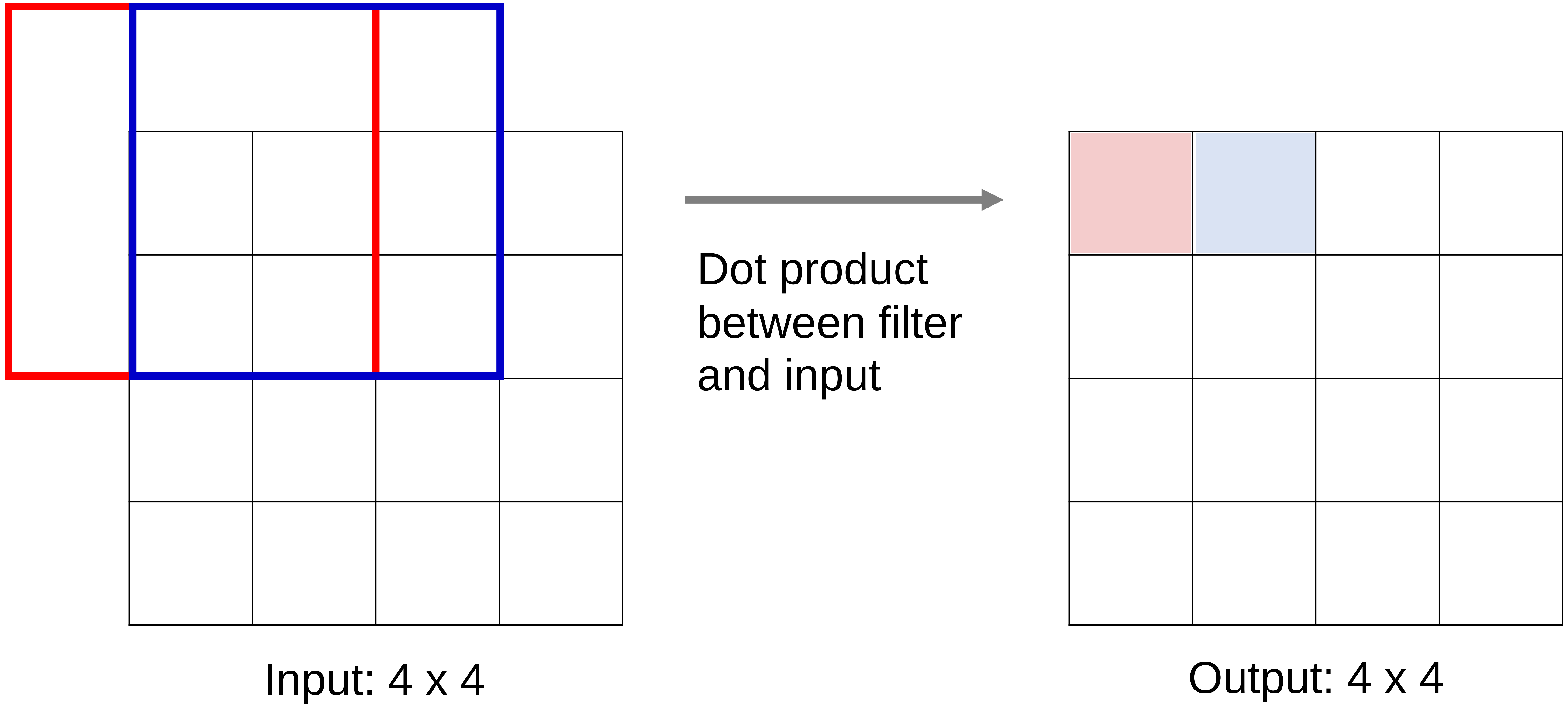
0	0	2	0
0	1	0	0
0	0	0	0
3	0	0	4

Output: 4 x 4

In max unpooling, the position of the maximum value in each pooling operation is recorded and used during the unpooling operation to place the value back into its original location in the feature map.

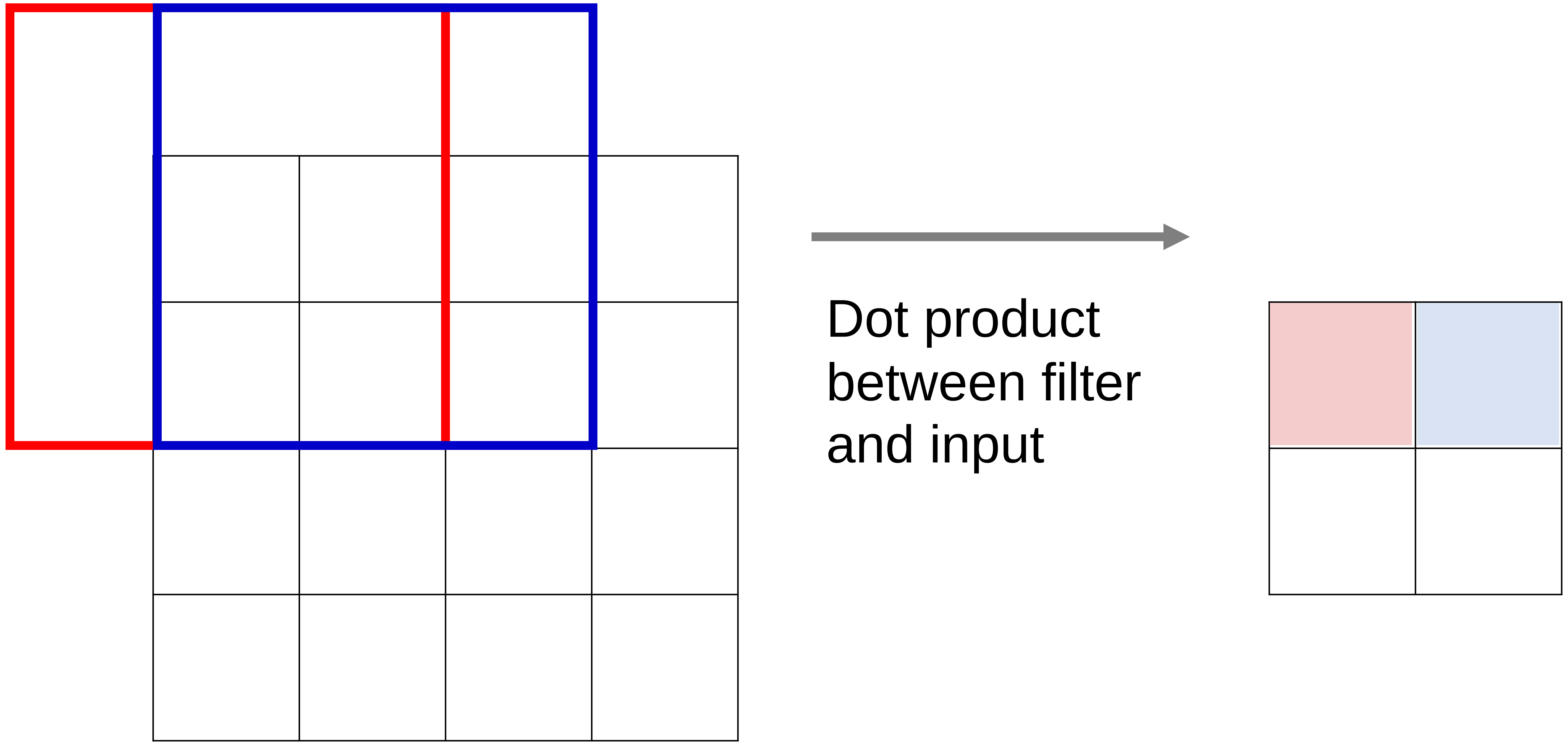
Learnable Upsampling

Recall: Normal 3 x 3 convolution, stride 1 pad 1



Learnable Upsampling

Recall: Normal 3 x 3 convolution, stride 2 pad 1



Input: 4 x 4

Output: 2 x 2

Filter moves 2 pixels in the input for every one pixel in the output

Stride gives ratio between movement in input and output

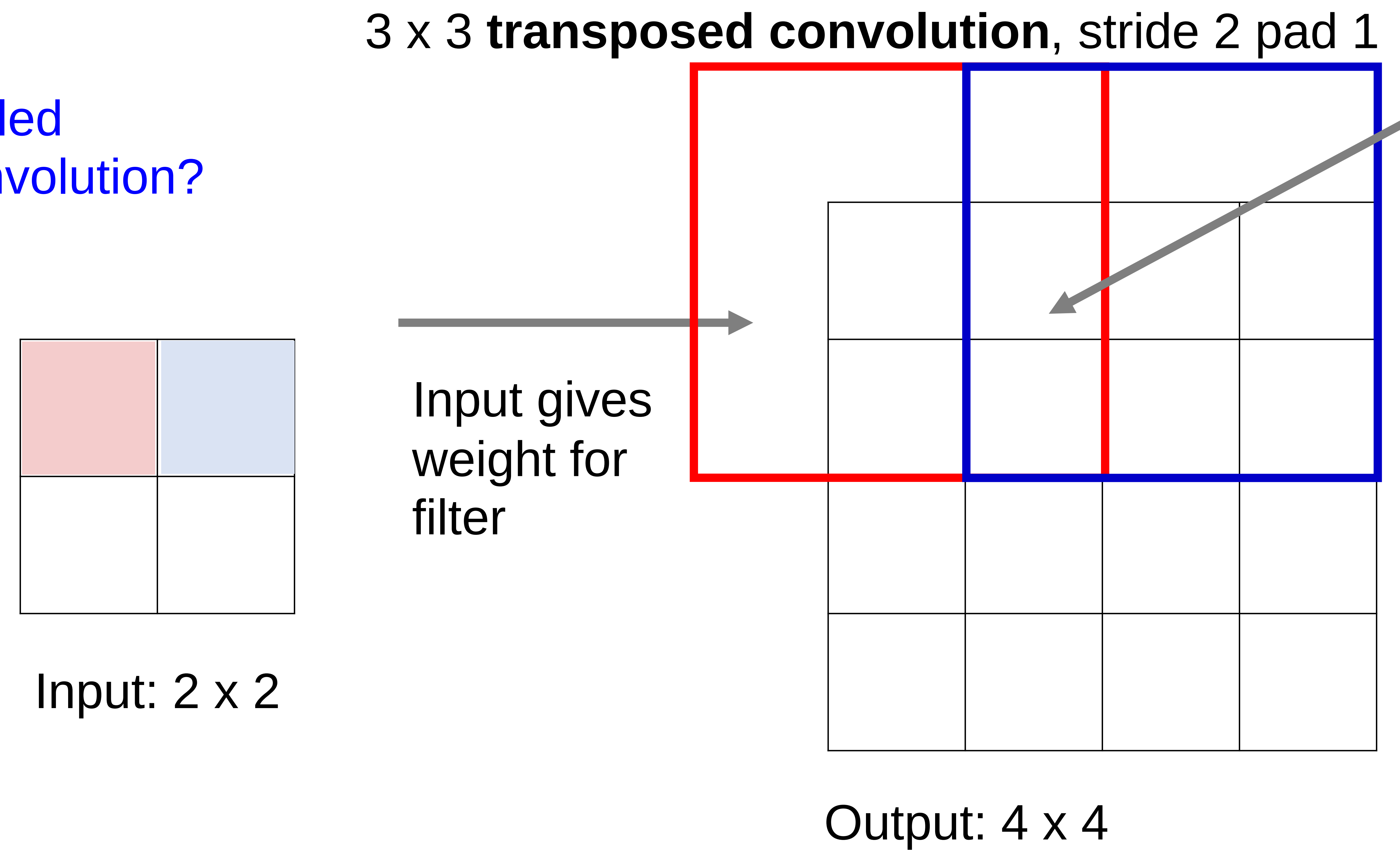
We can interpret strided convolution as “learnable downsampling”.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Learnable Upsampling: *Transposed Convolution*

Q: Why is it called transposed convolution?

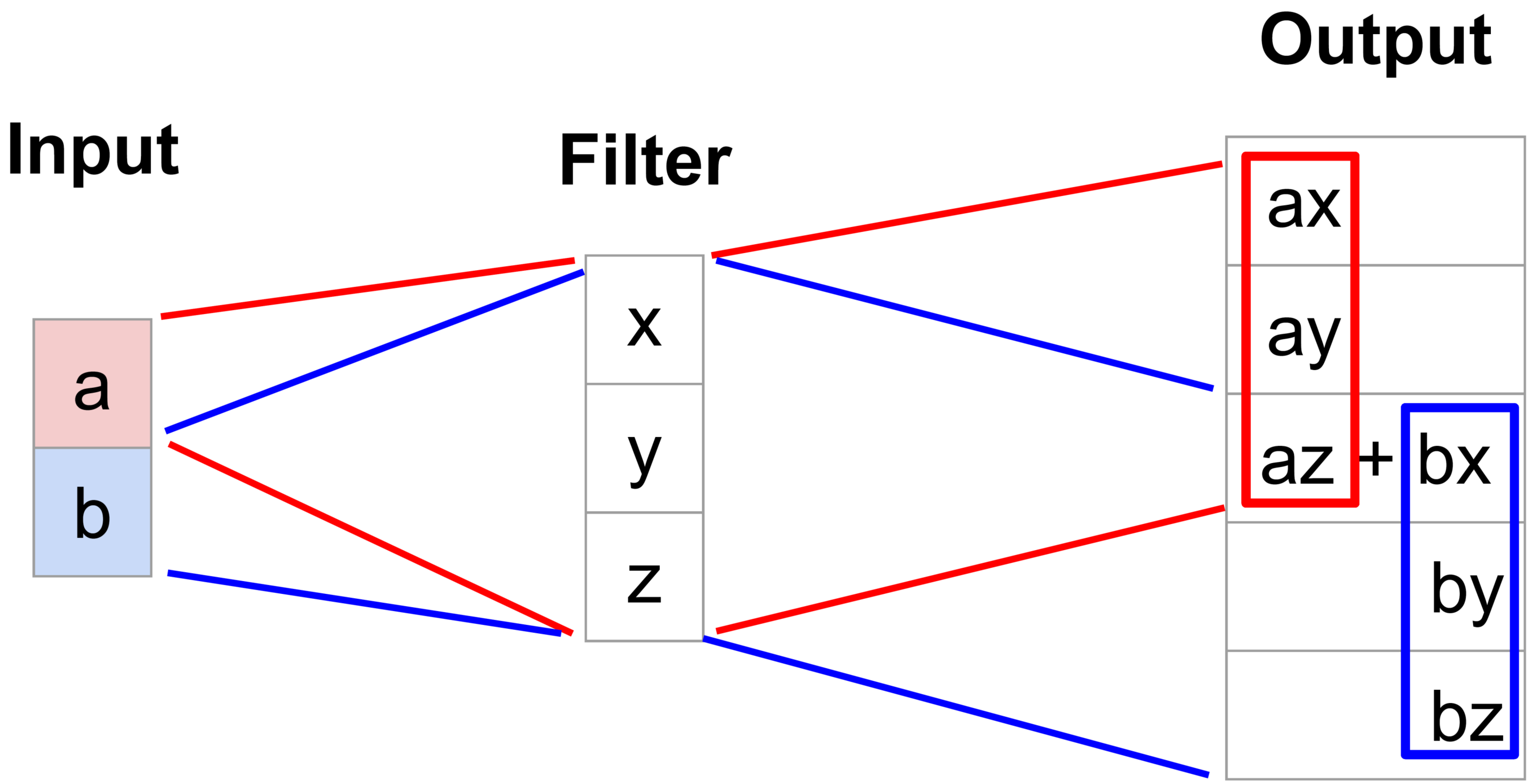


Sum where output overlaps

Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input

Learnable Upsampling: *1D Example*



Output contains copies of the filter weighted by the input, summing at where at overlaps in the output

Convolution as Matrix Multiplication (1D Example)

We can express convolution in terms of a matrix multiplication

$$\vec{x} * \vec{a} = X\vec{a}$$

$$\begin{bmatrix} x & y & z & 0 & 0 & 0 \\ 0 & 0 & x & y & z & 0 \end{bmatrix} \begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ay + bz \\ bx + cy + dz \end{bmatrix}$$

Example: 1D conv, kernel size=3, <u>stride=2</u>, padding=1

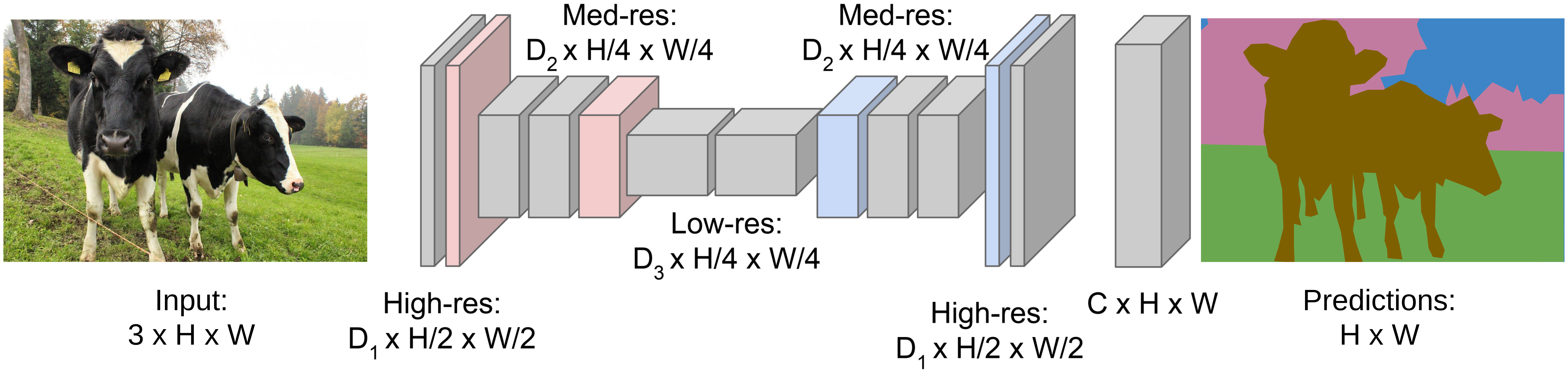
Transposed convolution multiplies by the transpose of the same matrix:

$$\vec{x} *^T \vec{a} = X^T\vec{a}$$

$$\begin{bmatrix} x & 0 \\ y & 0 \\ z & x \\ 0 & y \\ 0 & z \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} ax \\ ay \\ az + bx \\ by \\ bz \\ 0 \end{bmatrix}$$

Example: 1D transposed conv, kernel size=3, <u>stride=2</u>, padding=0

Semantic Segmentation Idea: *Fully Convolutional*



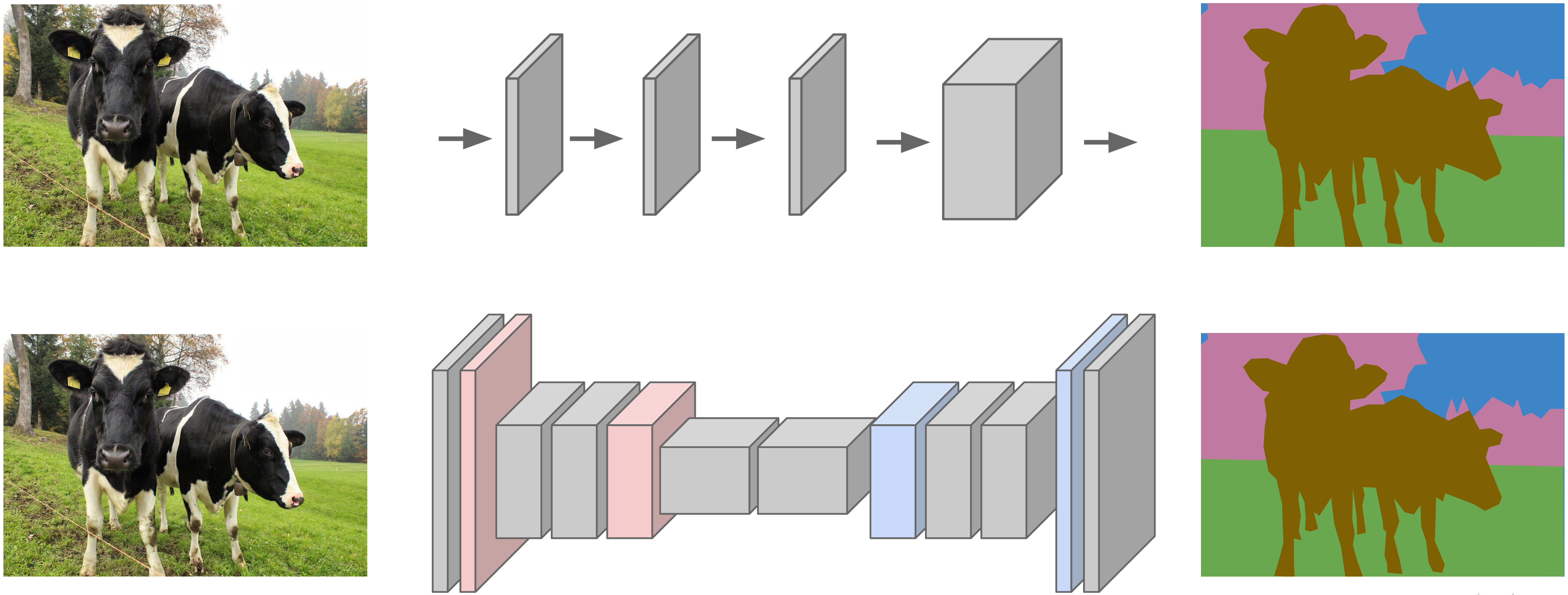
Downsampling:
Pooling, strided convolution

Unpooling or strided transposed convolution

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

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Semantic Segmentation: *Summary*



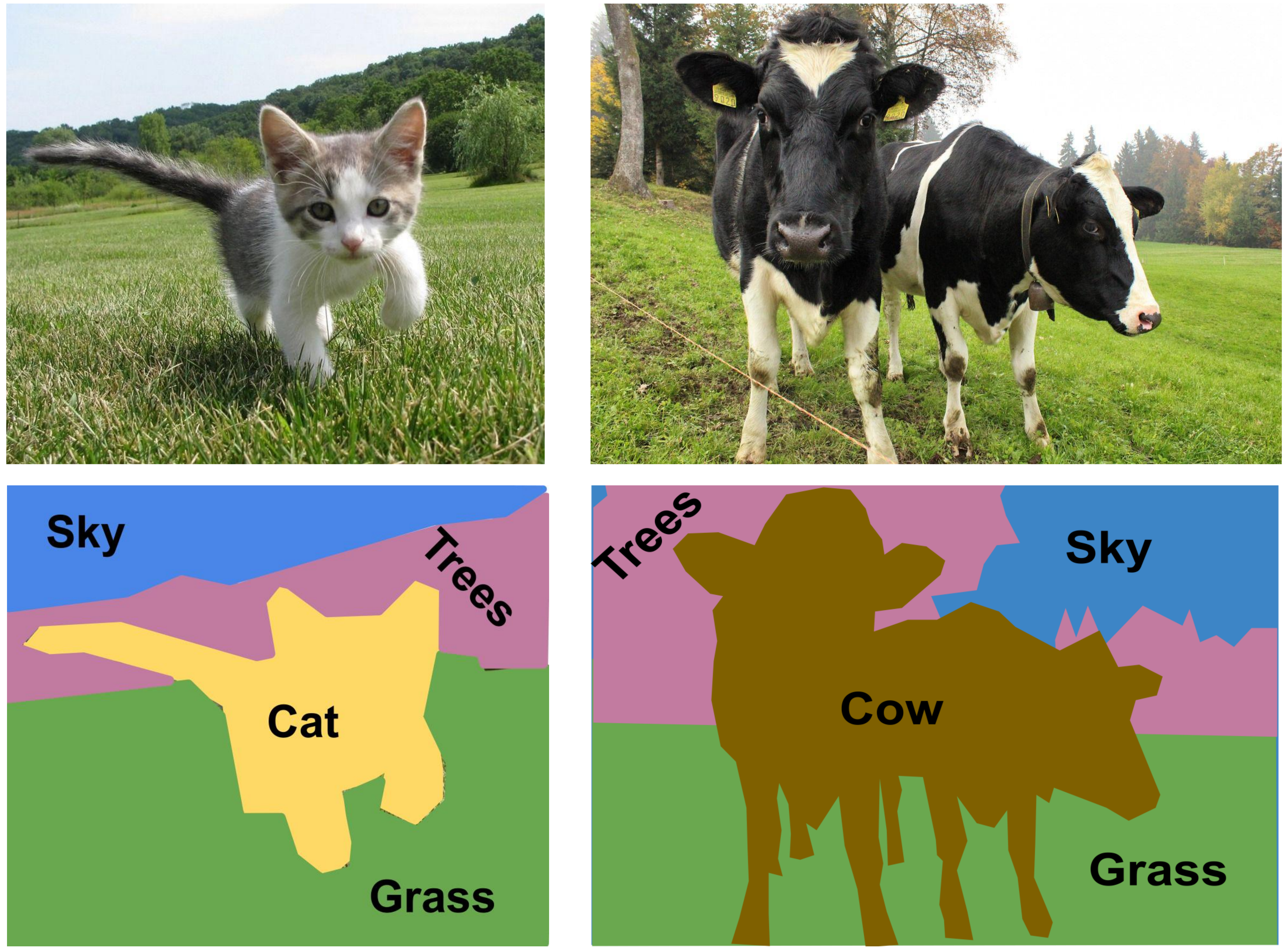
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Semantic Segmentation

Label each pixel in the image with a category label

Don’t differentiate instances, only care about pixels



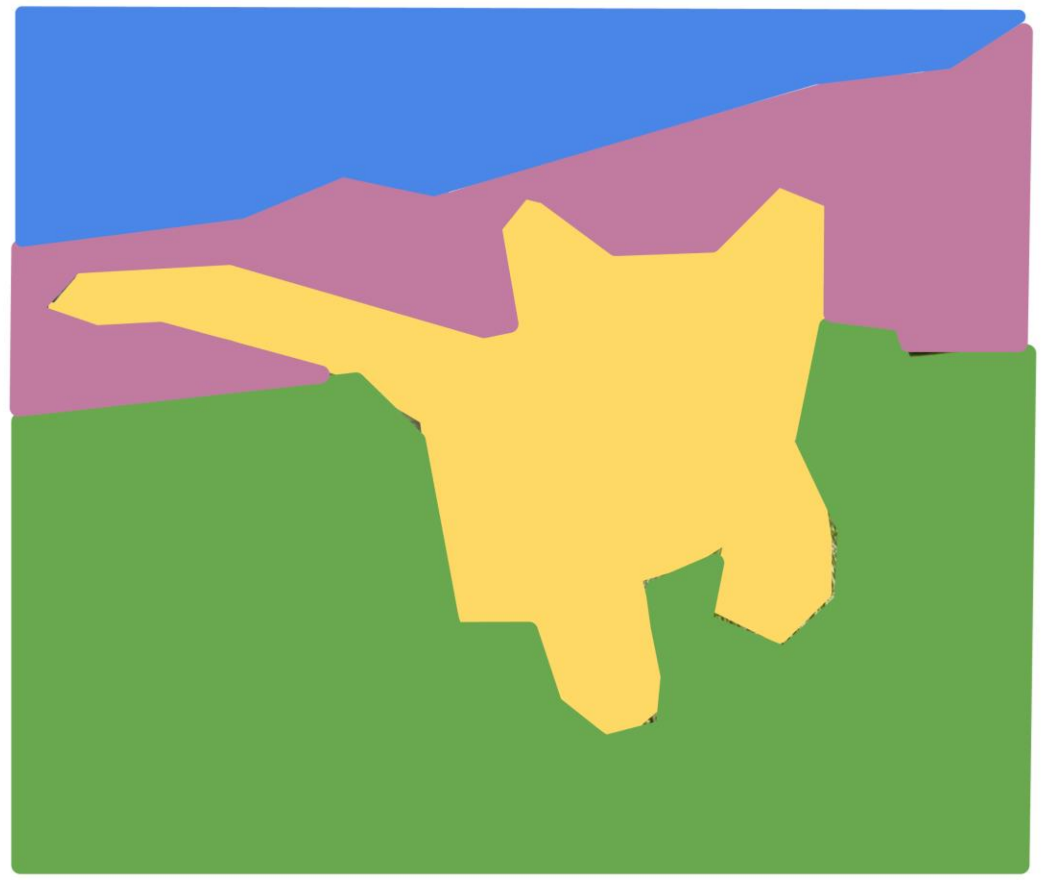
Computer Vision Tasks

Classification

CAT

No spatial extent

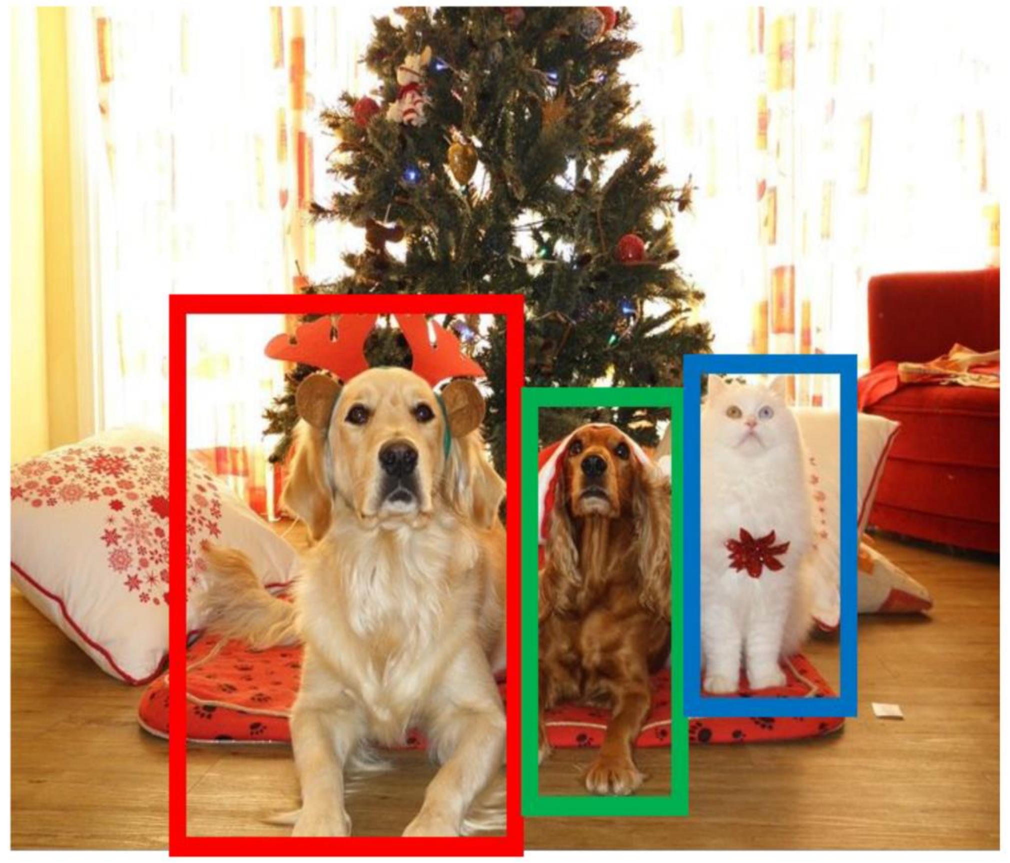
Semantic Segmentation



GRASS, CAT, TREE, SKY

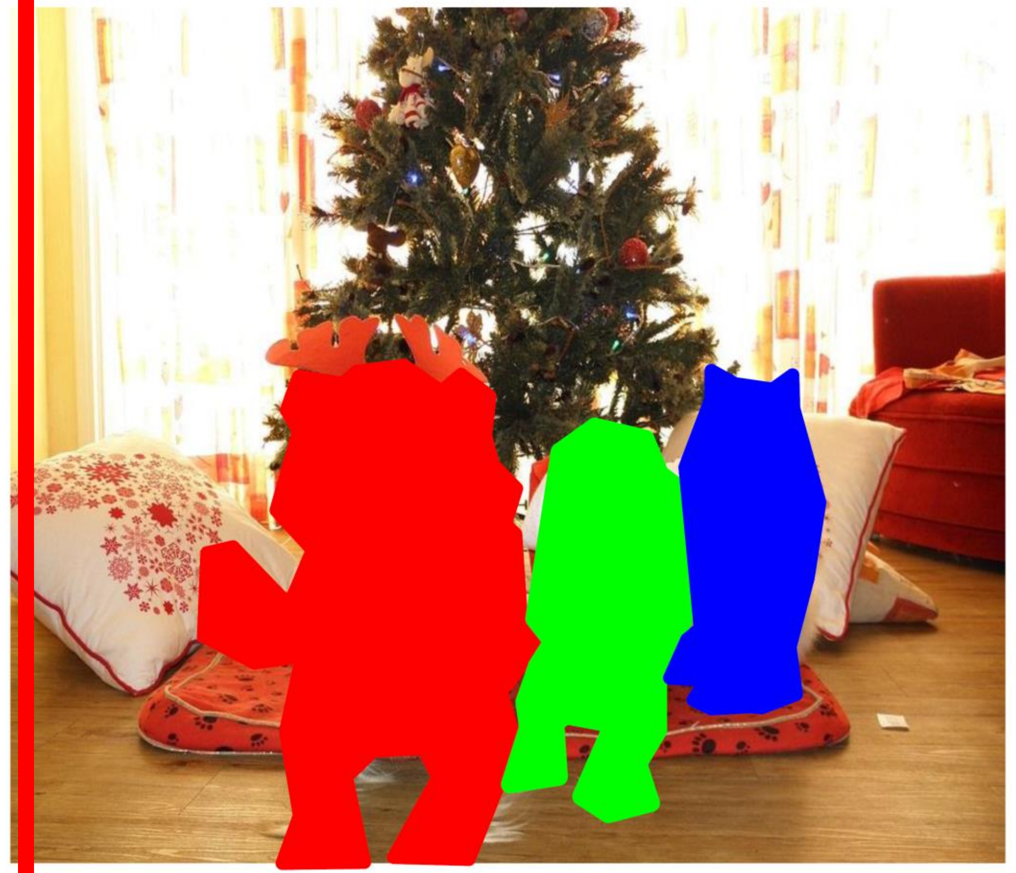
No objects, just pixels

Object Detection



DOG, DOG, CAT

Instance Segmentation

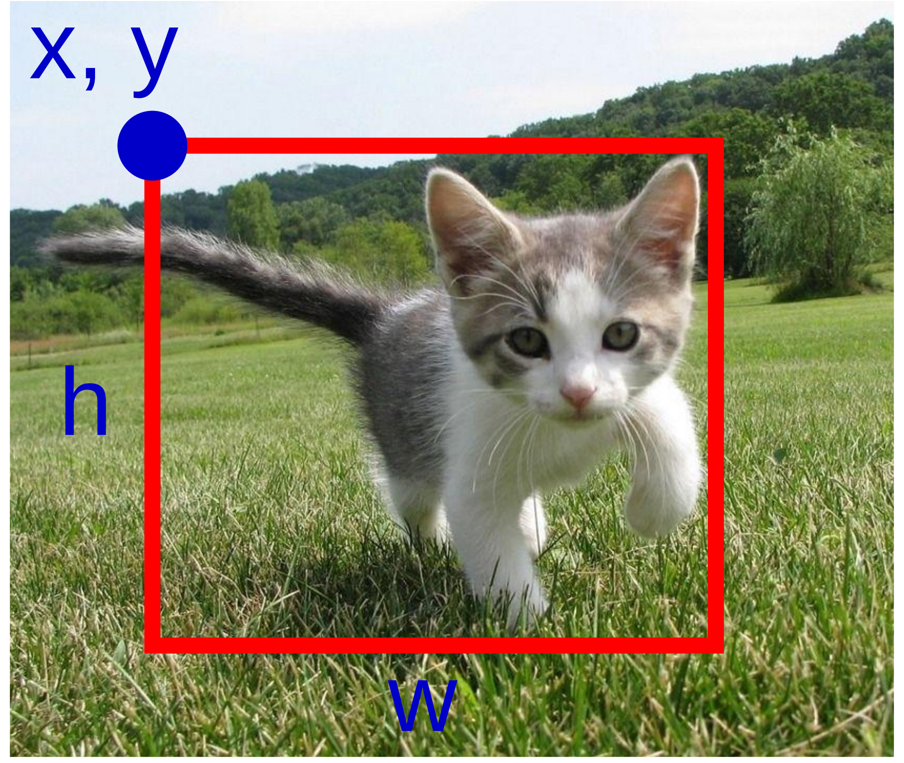


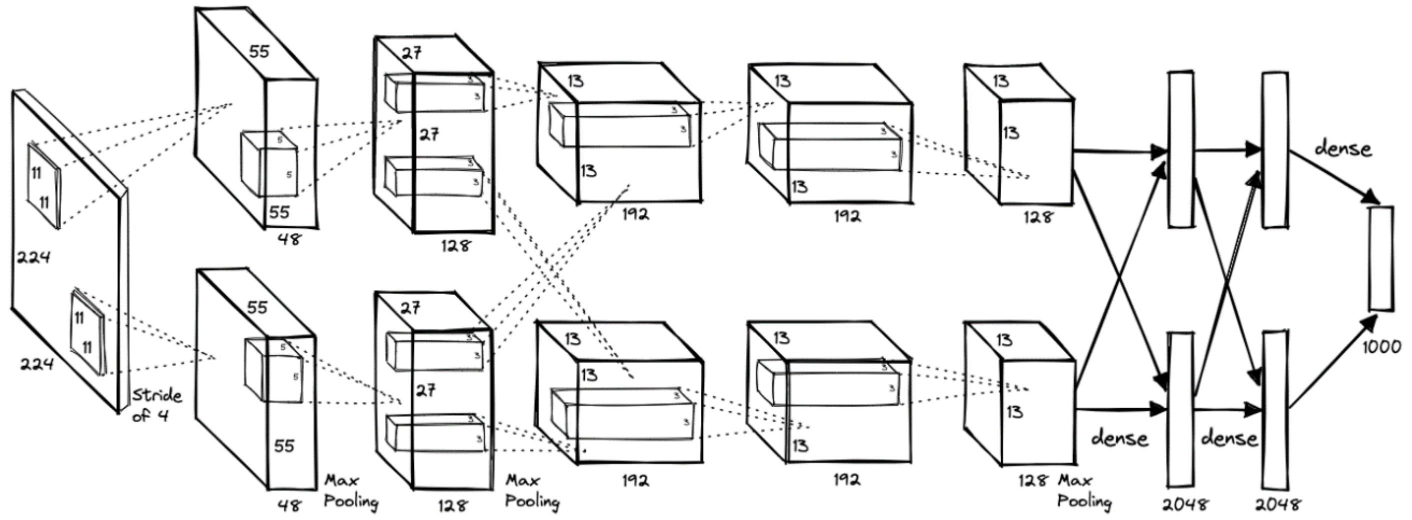
DOG, DOG, CAT

Multiple Object

This image is CC0 public domain

Object Detection: *Single Object (Classification + Localization)*





Vector: 4096

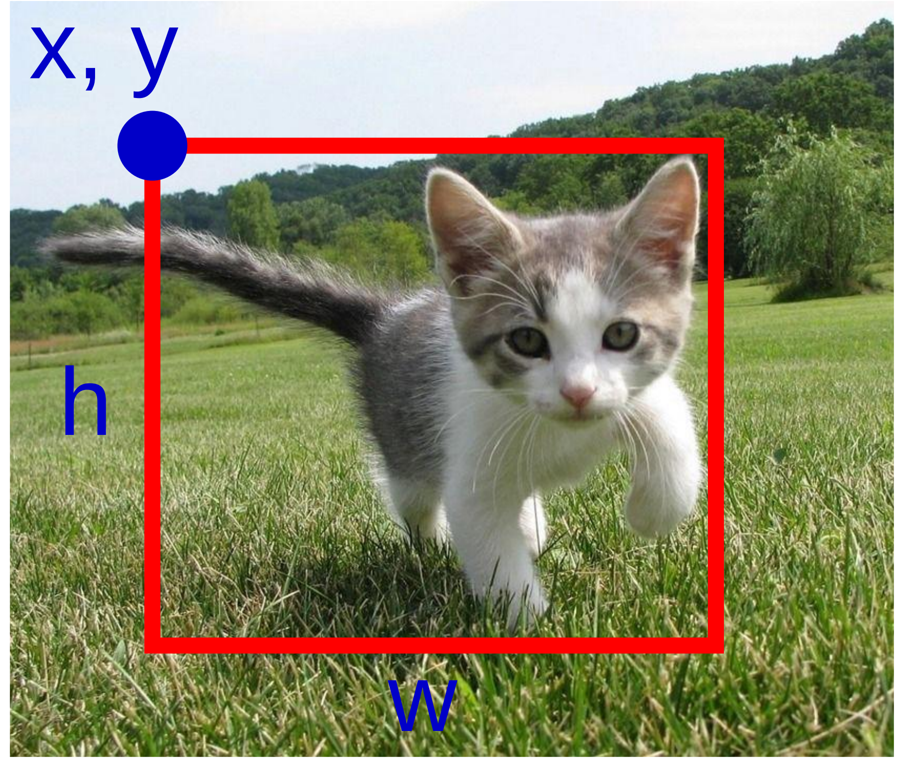
Fully Connected: 4096 to 1000

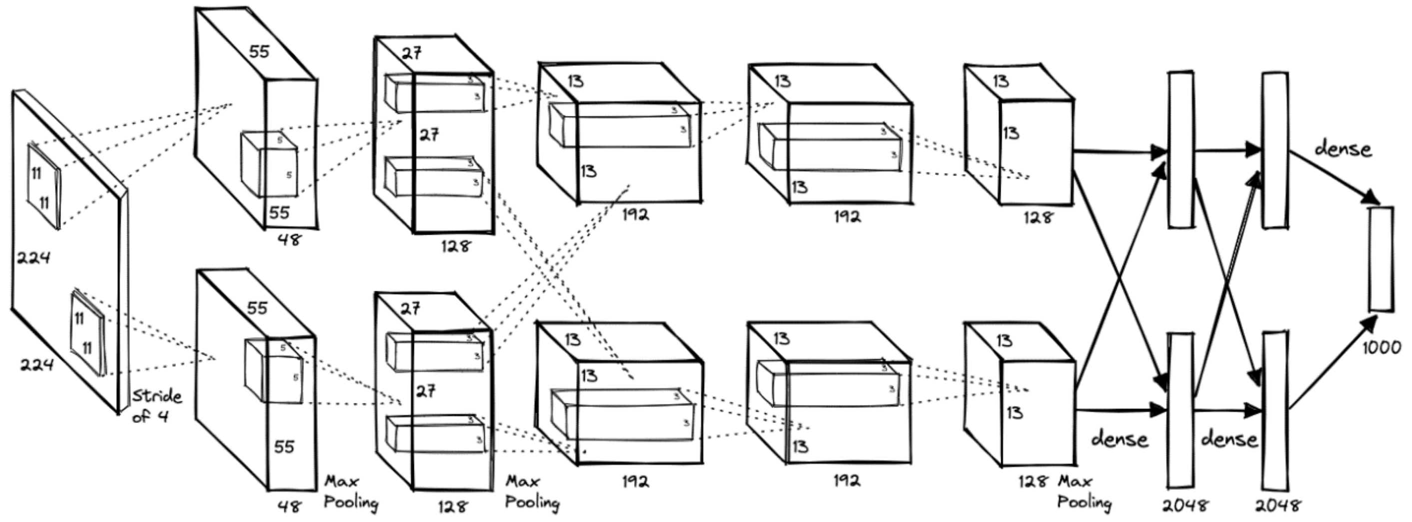
Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Fully Connected: 4096 to 4

Box Coordinates
(x, y, w, h)

Object Detection: *Single Object (Classification + Localization)*





Treat localization as a regression problem!

Vector: 4096

Fully Connected: 4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Softmax Loss

Correct label: Cat

Fully Connected: 4096 to 4

Box Coordinates (x, y, w, h)

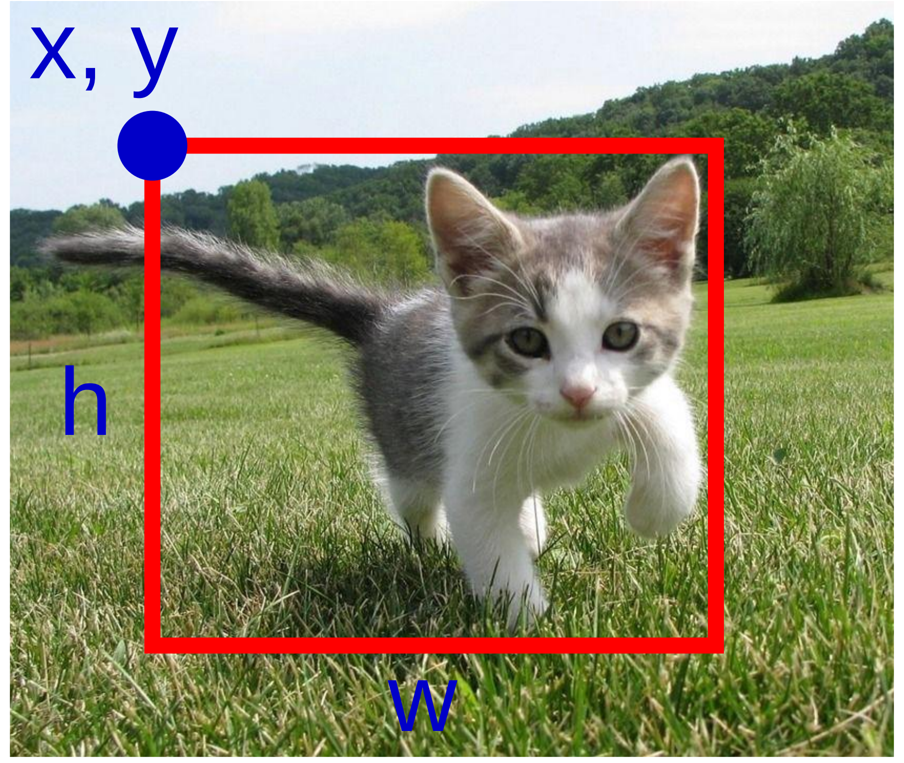
L2 Loss

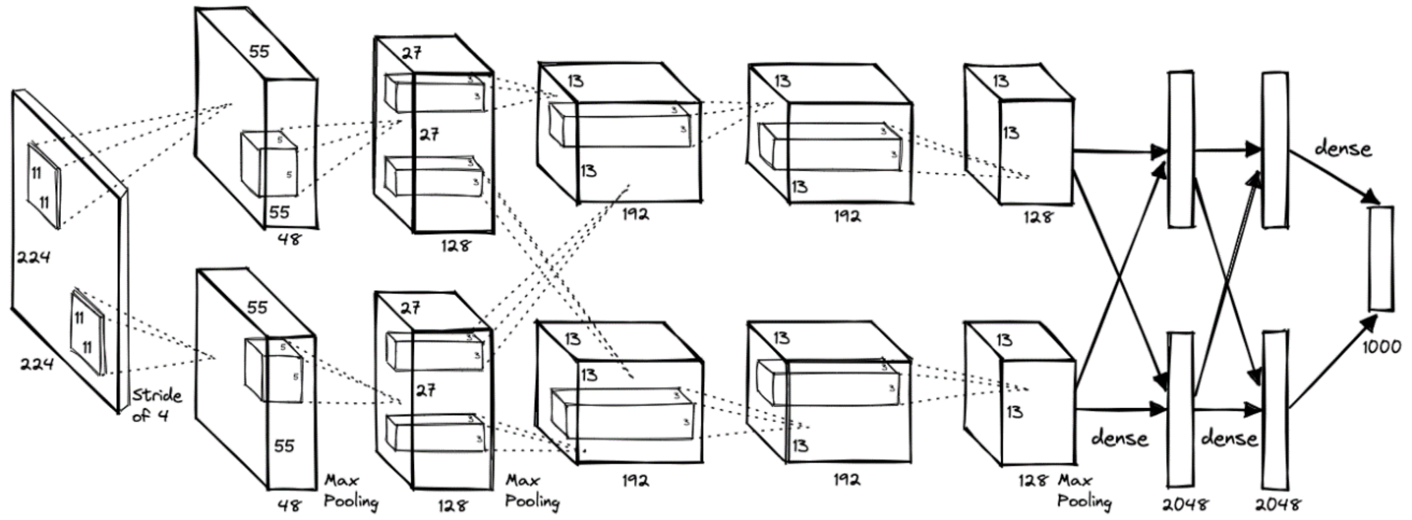
Correct box: (x’, y’, w’, h’)

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Object Detection: *Single Object (Classification + Localization)*





Treat localization as a regression problem!

Vector: 4096

Fully Connected: 4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Correct label: Cat

Softmax Loss

Multitask Loss

+ → **Loss**

Fully Connected: 4096 to 4

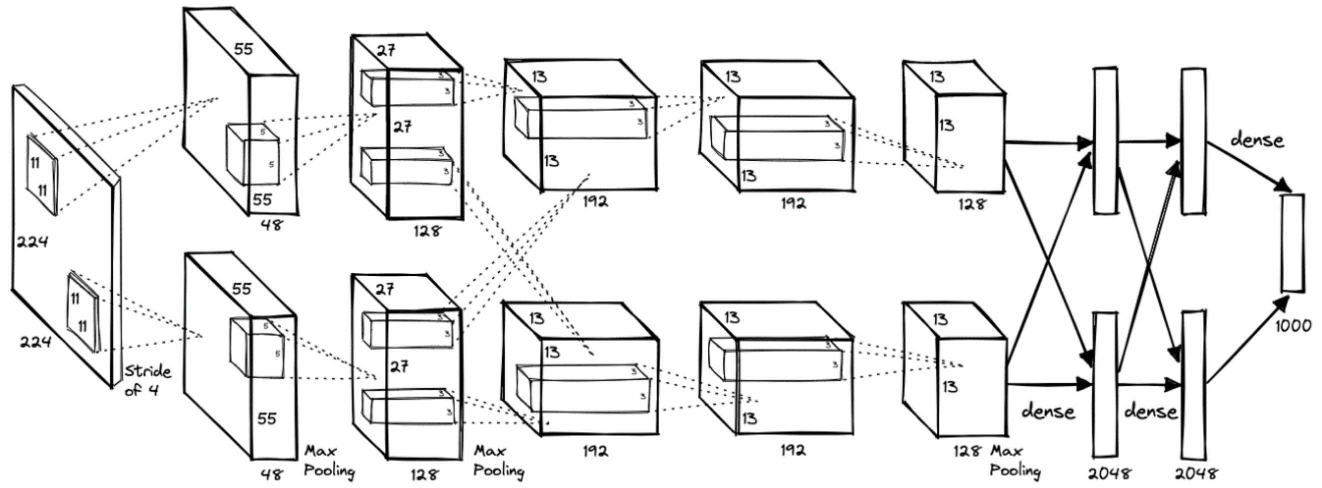
Box Coordinates (x, y, w, h)

L2 Loss

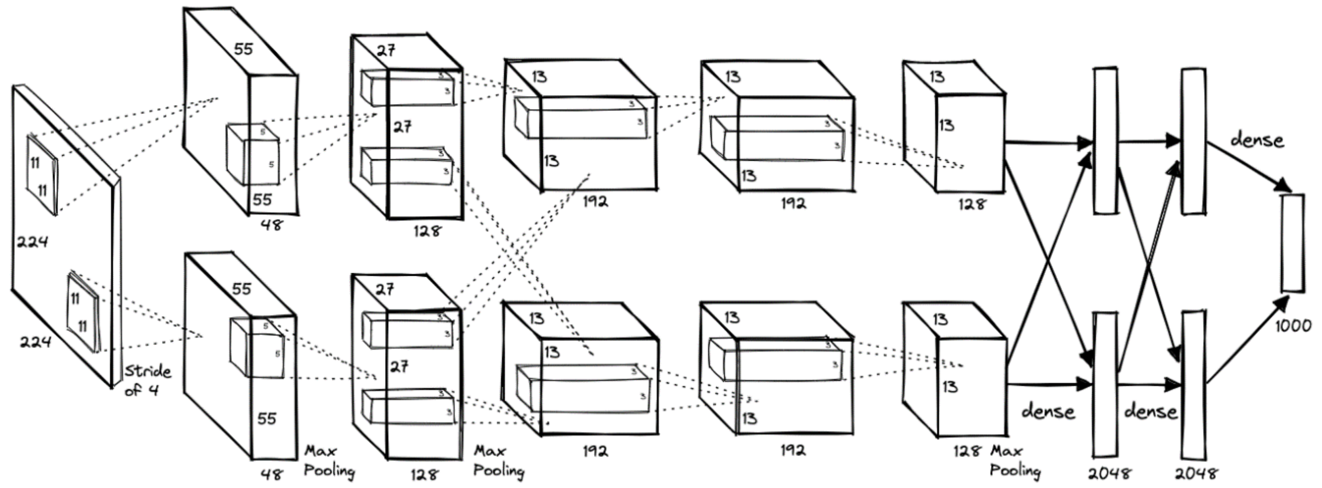
Correct box: (x’, y’, w’, h’)

Object Detection: *Multiple Objects*

Each image needs a different number of outputs!

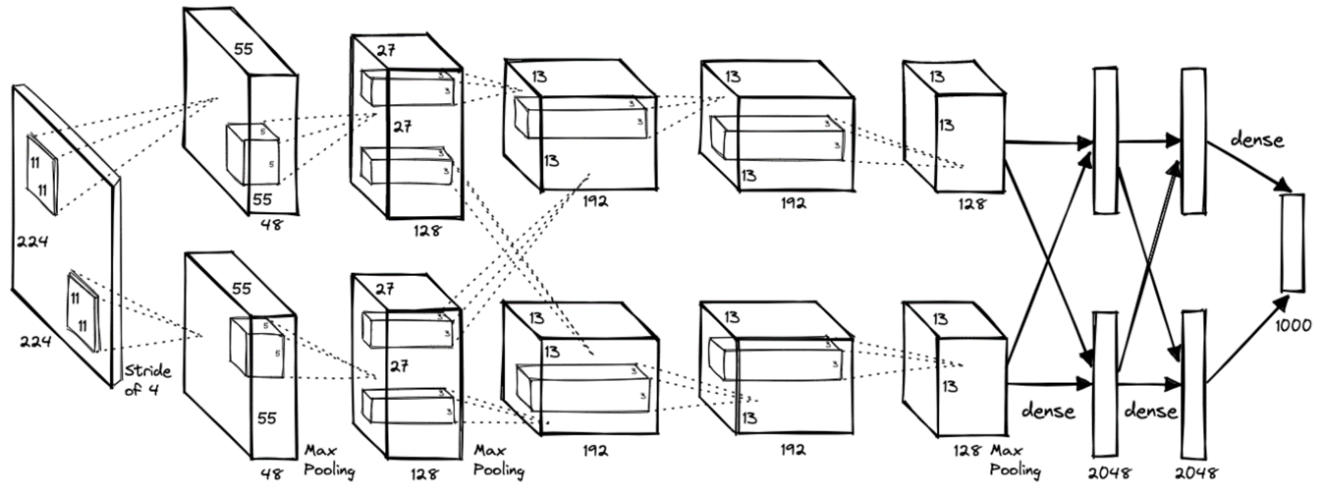



CAT: (x, y, w, h) 4 numbers

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

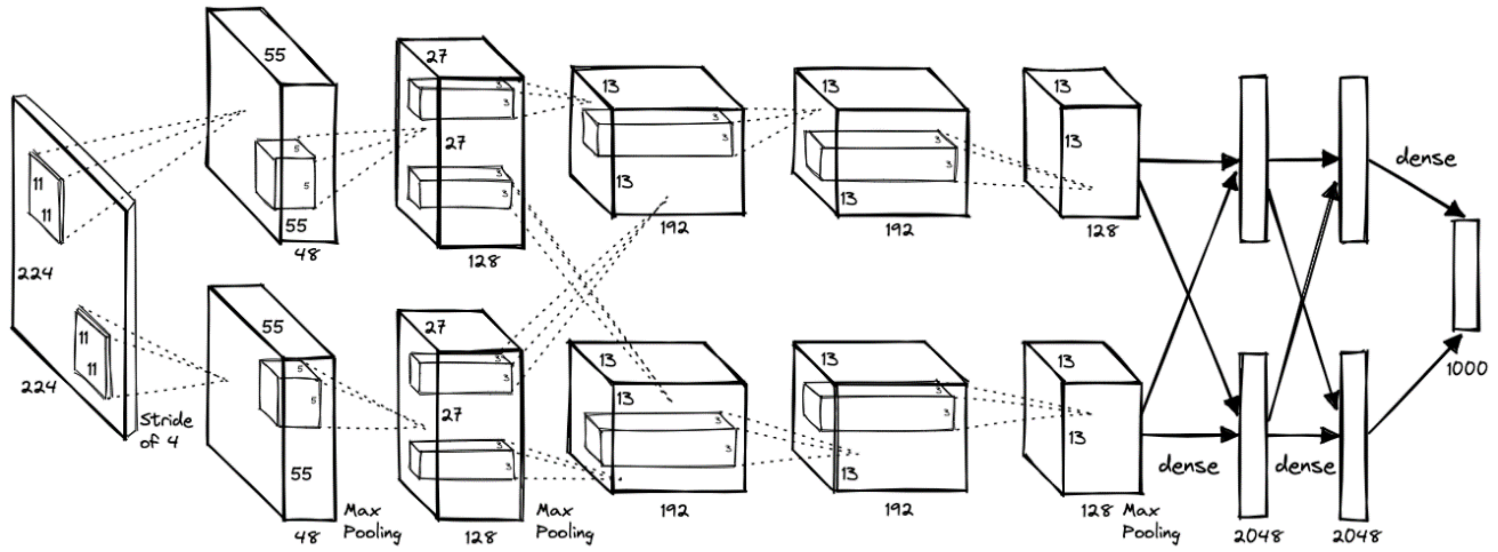
12 numbers

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

Many numbers!

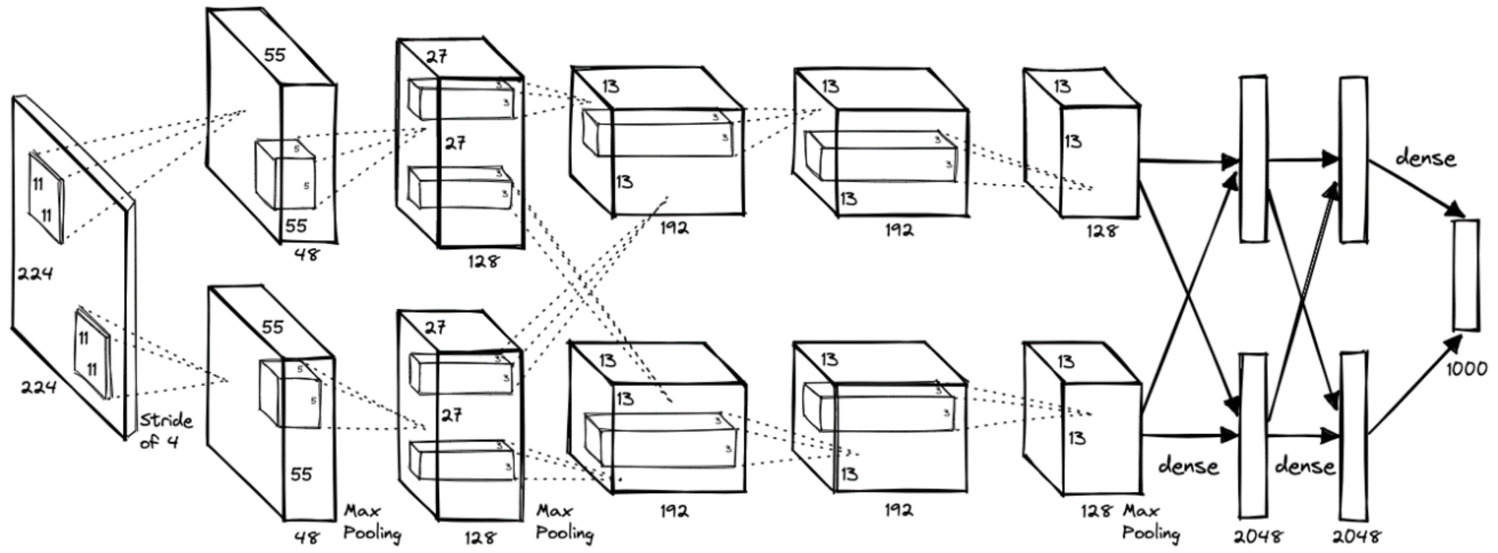
Object Detection: *Multiple Objects*



Dog? NO
Cat? NO
Background? YES

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

Object Detection: *Multiple Objects*

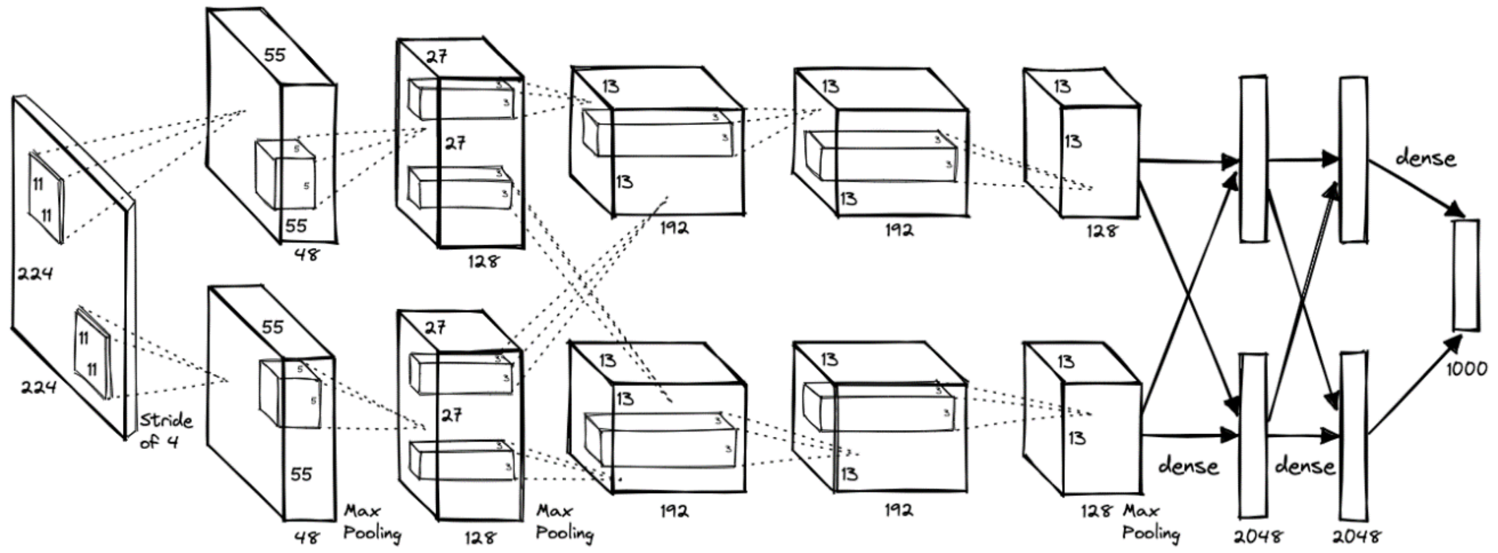


Dog? YES
Cat? NO
Background? NO

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

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Object Detection: *Multiple Objects*

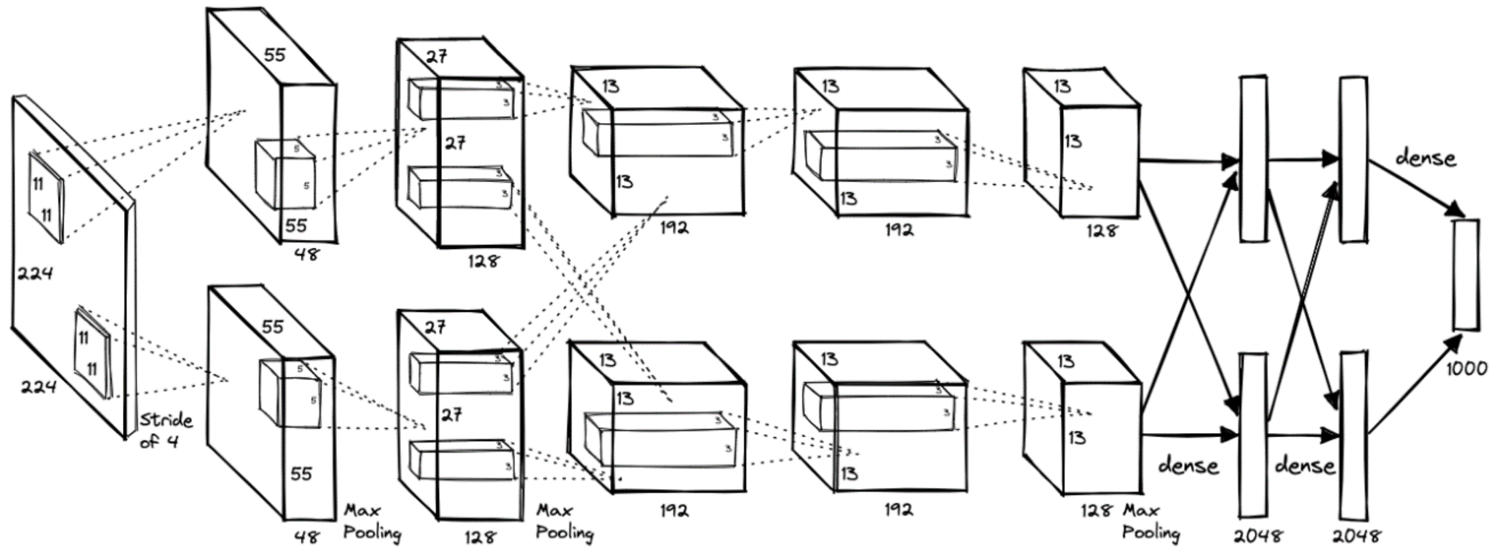


Dog? YES
Cat? NO
Background? NO

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

Object Detection: *Multiple Objects*

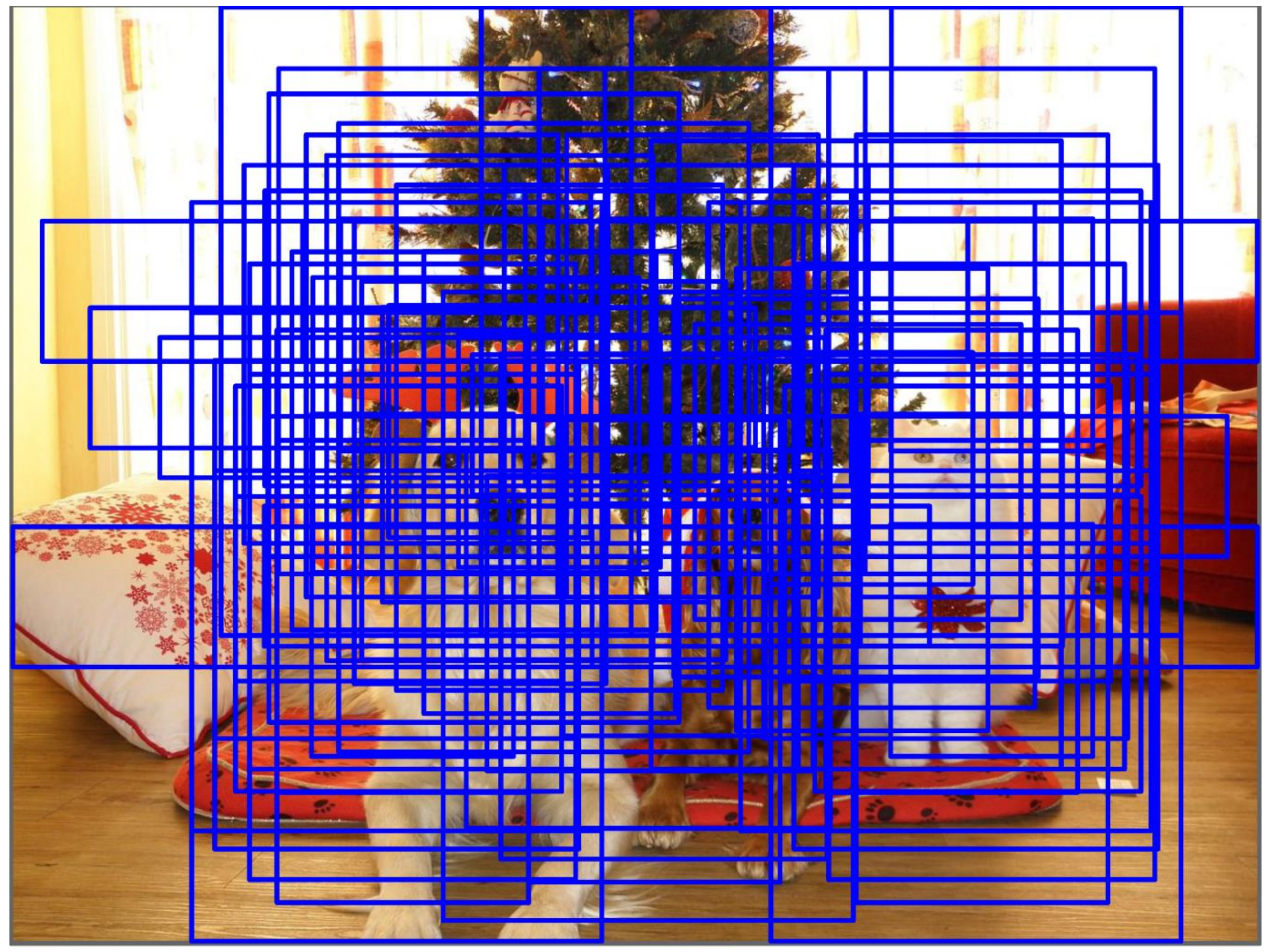
Q: What's the problem with this approach?



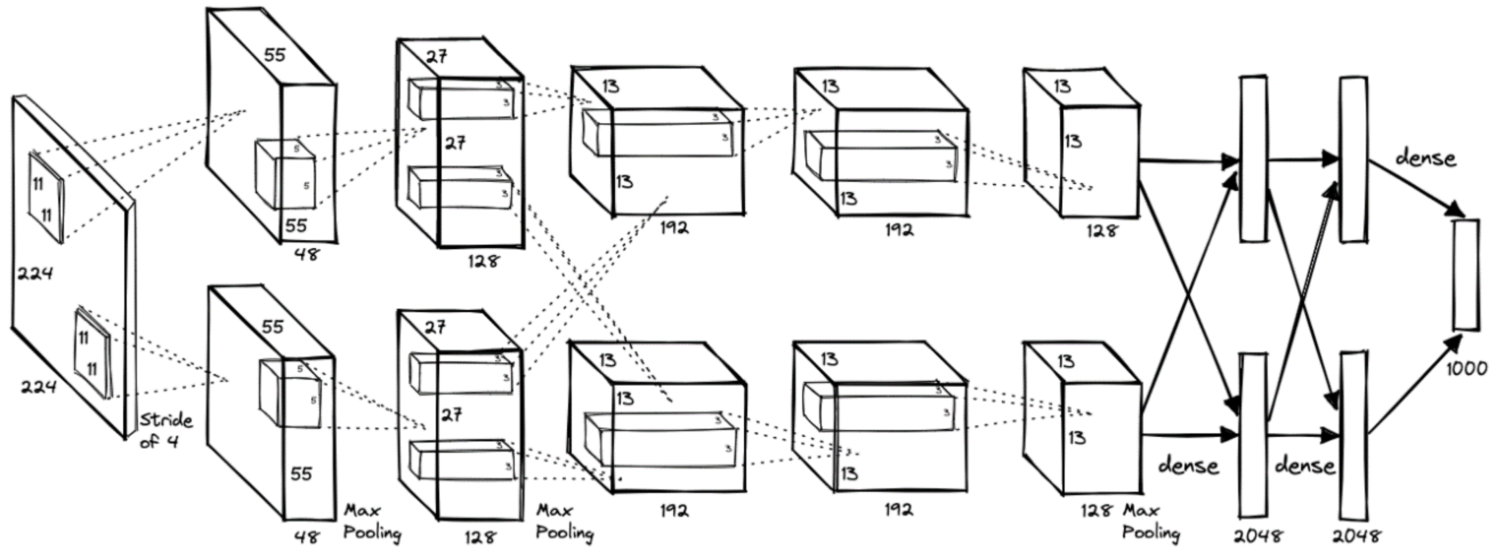
Dog? NO
Cat? YES
Background? NO

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

Object Detection: *Multiple Objects*



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



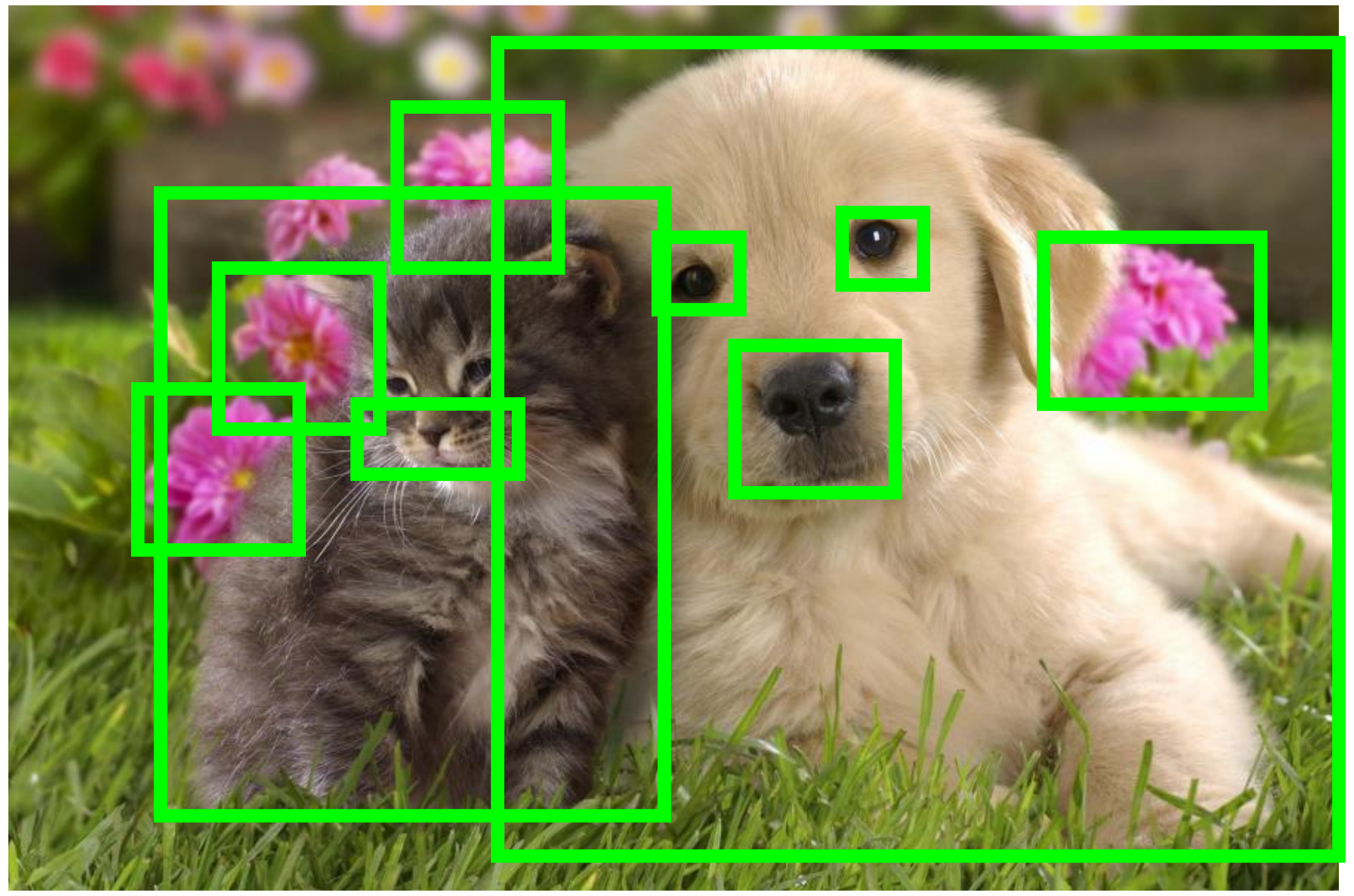
Dog? NO
Cat? YES
Background? NO

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

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Region Proposals: *Selective Search*



- Find “blobby” 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

Alexe et al, “Measuring the objectness of image windows”, TPAMI 2012
Uijlings et al, “Selective Search for Object Recognition”, IJCV 2013
Cheng et al, “BING: Binarized normed gradients for objectness estimation at 300fps”, CVPR 2014
Zitnick and Dollar, “Edge boxes: Locating object proposals from edges”, ECCV 2014

R-CNN (Region-based Convolutional Neural Network)

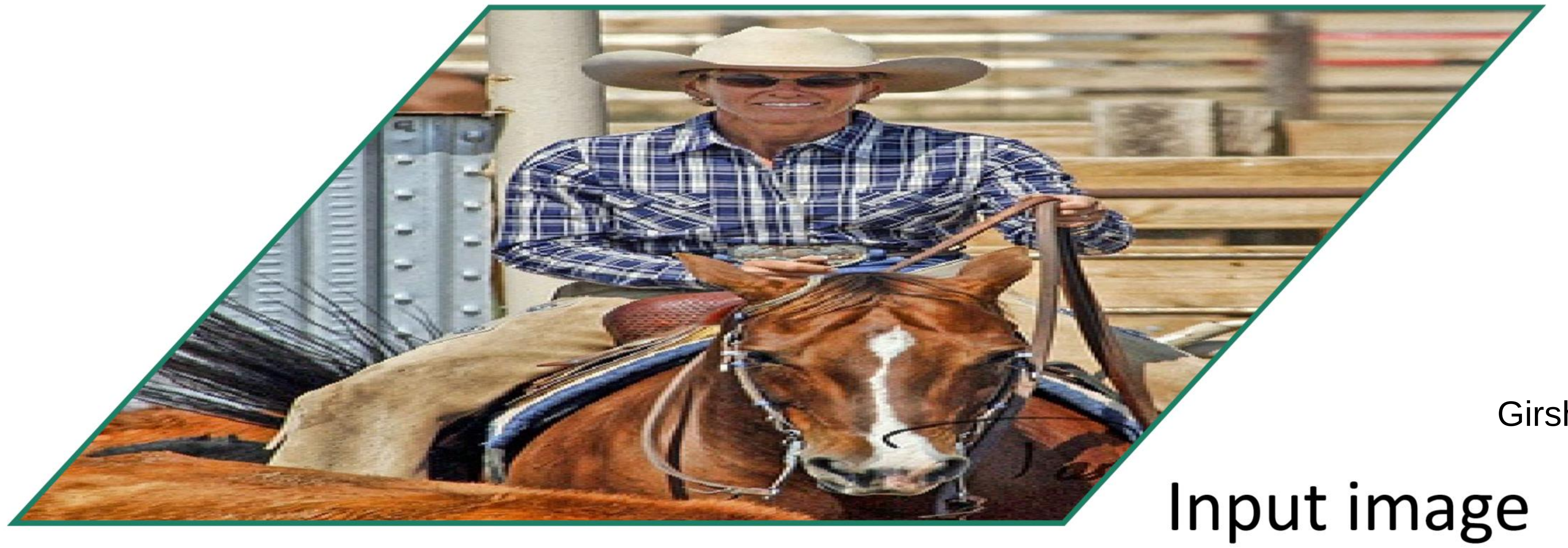
R-CNN (Region-based Convolutional Neural Network) is a popular object detection framework that consists of three main components: region proposal, feature extraction, and object classification.

Here's how R-CNN works:

1. **Region Proposal:** First, the image is divided into regions using a selective search algorithm. The algorithm analyzes the image at multiple scales and identifies regions that are likely to contain objects.
2. **Feature Extraction:** Next, a convolutional neural network (CNN) is used to extract a fixed-length feature vector from each region proposal. The CNN is typically pre-trained on a large dataset such as ImageNet to learn a generic set of features that can be used for a wide range of object detection tasks.
3. **Object Classification:** Finally, a set of support vector machines (SVMs) are trained to classify the extracted features into different object categories. A separate SVM is trained for each object category of interest.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

R-CNN



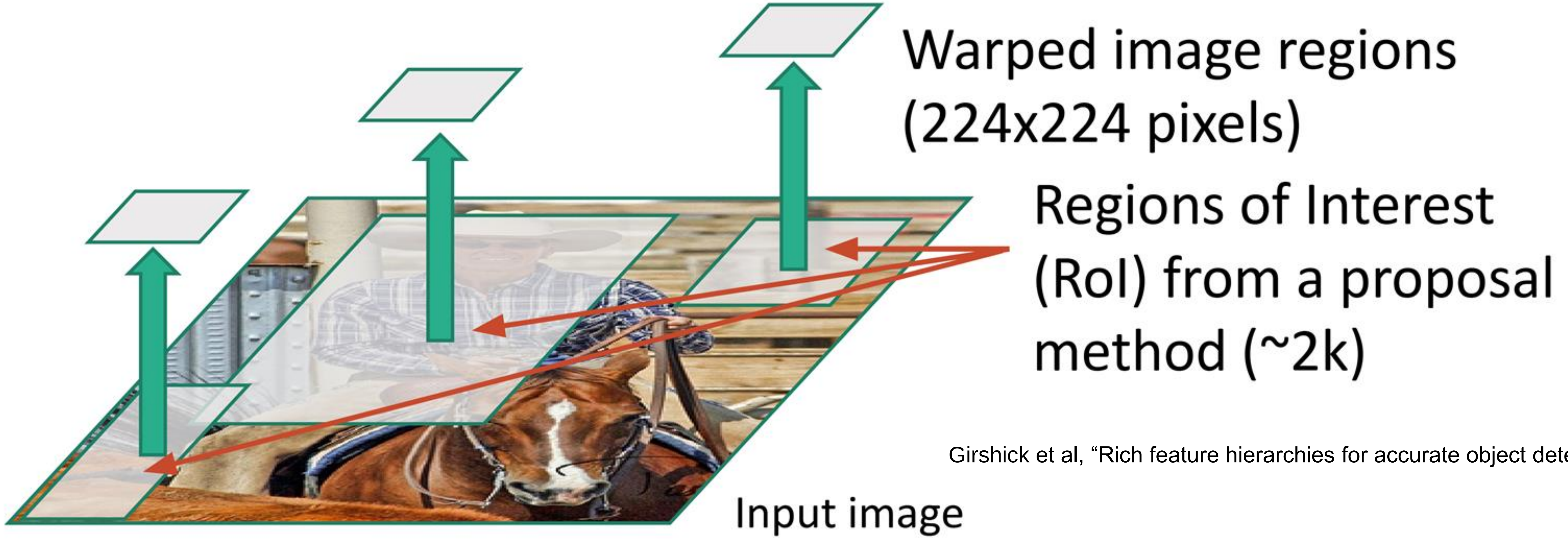
Input image

Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.

R-CNN

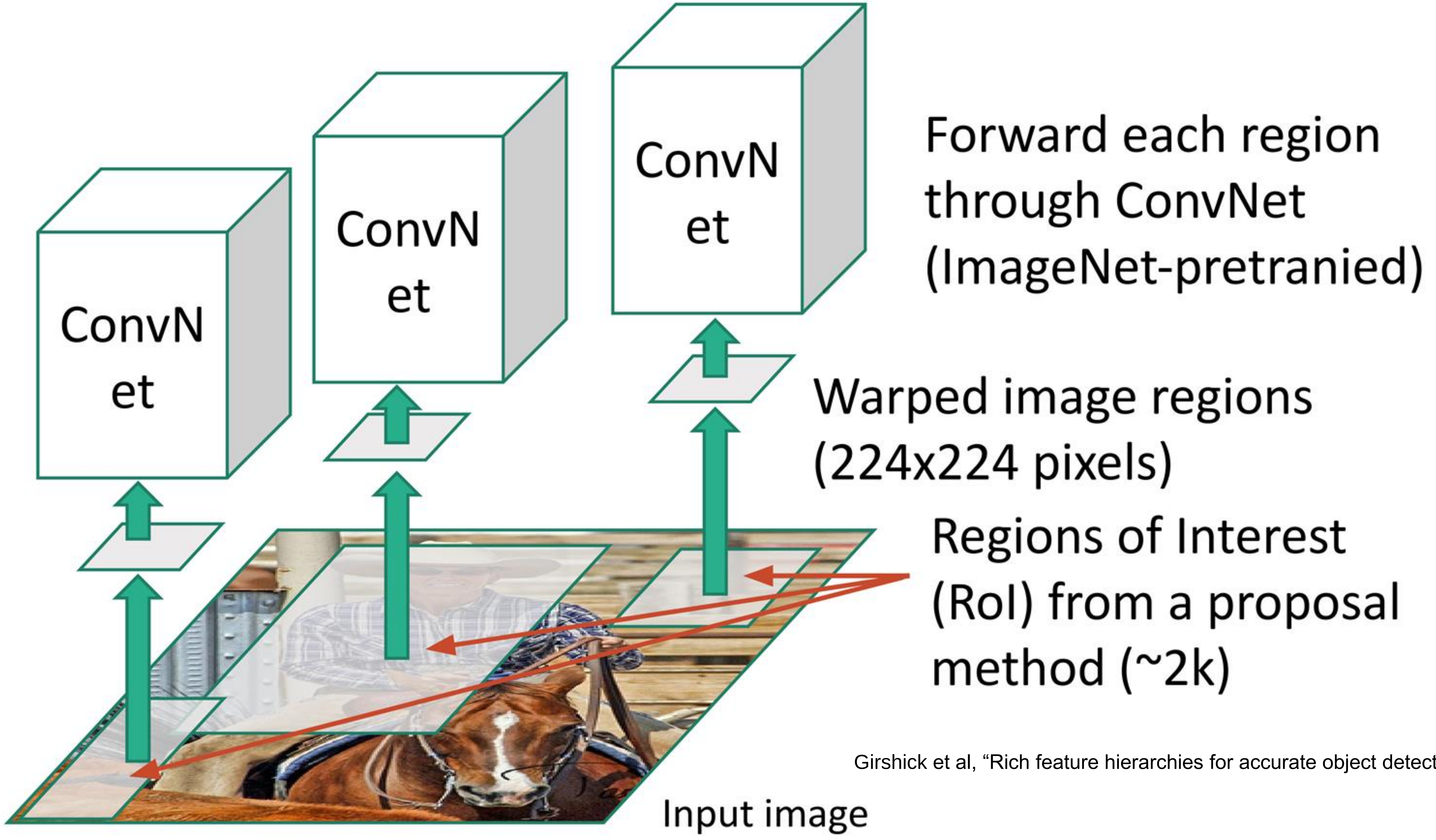
Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.

R-CNN



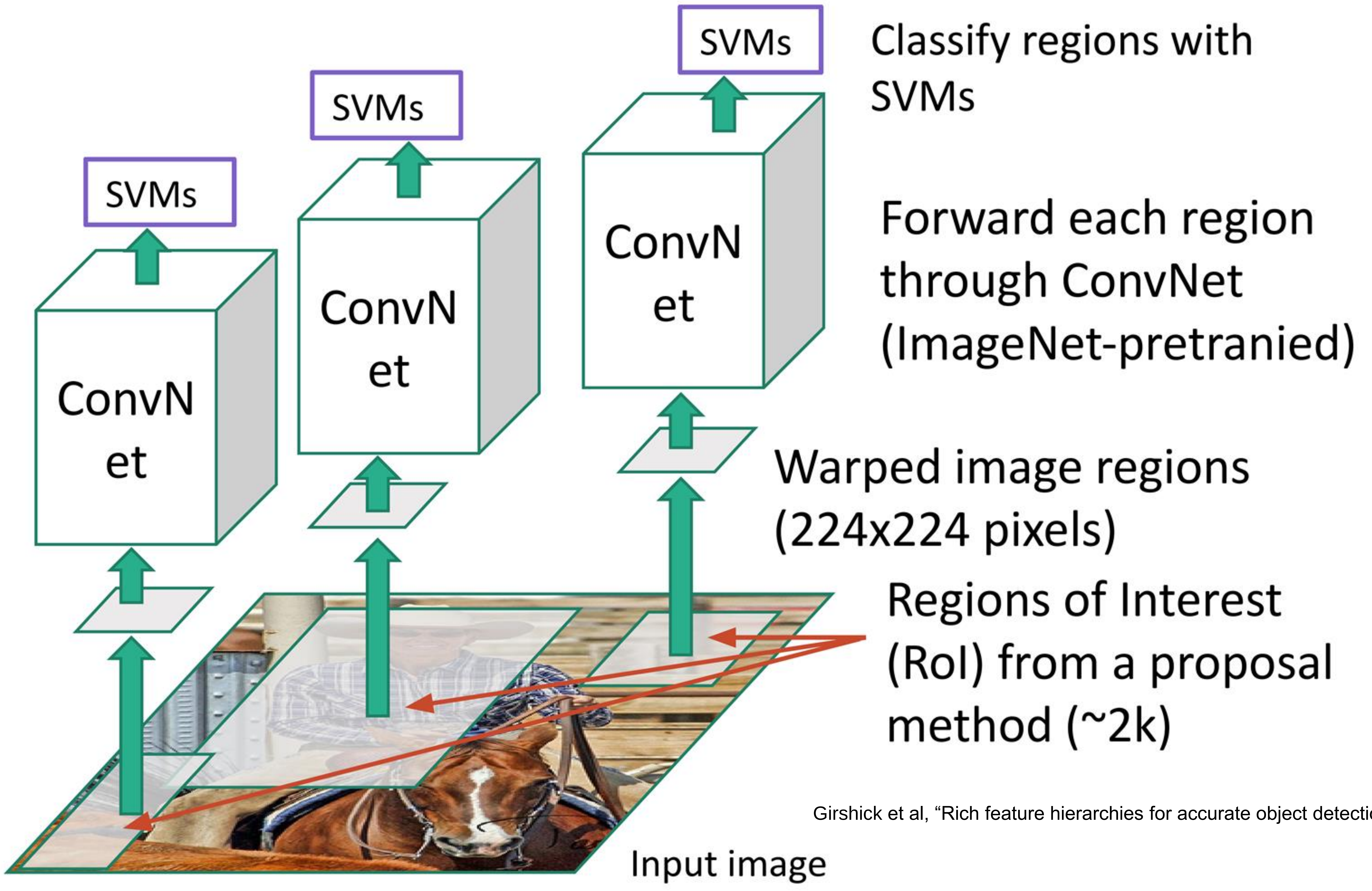
Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.

R-CNN



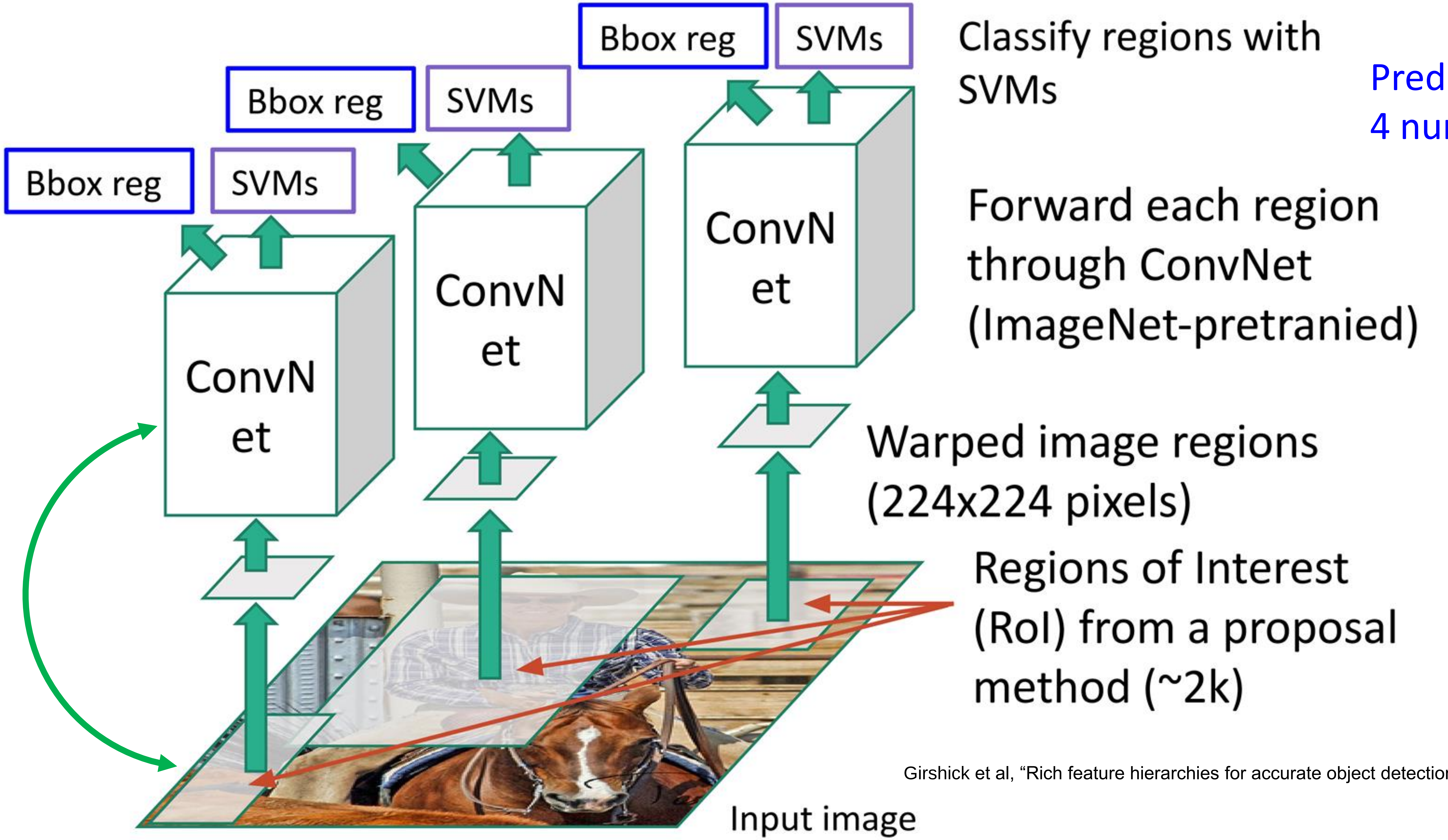
Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.

R-CNN



Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.

R-CNN



Predict "corrections" to the RoI: 4 numbers: (dx, dy, dw, dh)

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!

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

R-CNN: *Limitations*

1. **Slow Training and Inference:** R-CNN is a slow method since it requires multiple stages to detect objects. The selective search algorithm used for region proposal is computationally expensive, and the CNN feature extraction also takes a lot of time. This makes R-CNN unsuitable for real-time object detection applications.

2. **High Memory Usage:** R-CNN requires a lot of memory to store the intermediate results from the selective search algorithm and the CNN feature extraction. This can be a problem when processing high-resolution images or when running on devices with limited memory.

3. **Non-End-to-End Training:** The training of R-CNN is not end-to-end, which means that the different components are trained independently. This can lead to suboptimal performance since the different components may not be optimized for the overall task.

4. **Difficulty Handling Overlapping Objects:** R-CNN can struggle with detecting overlapping objects since the selective search algorithm tends to generate a lot of proposals, and the CNN feature extraction may not be able to distinguish between different objects that are close together.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Fast R-CNN

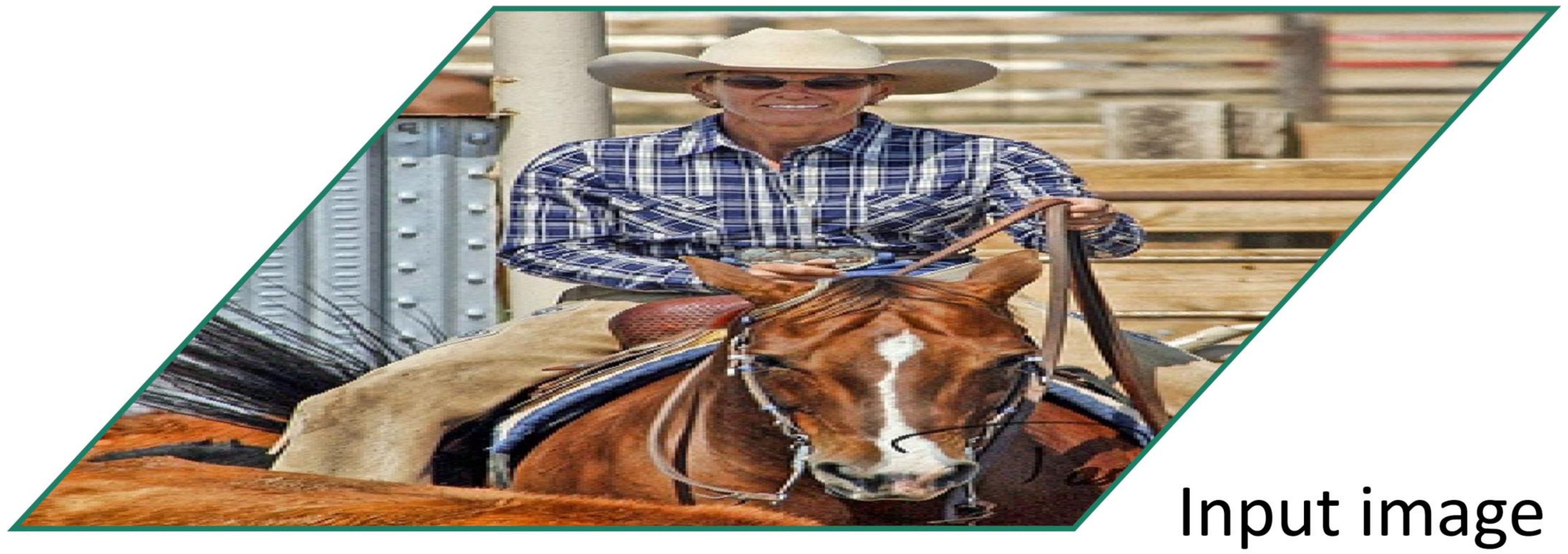
Fast R-CNN is an improvement over the original R-CNN framework that addresses some of its limitations, particularly its slow training and inference times. Here's how Fast R-CNN works:

1. **Region Proposal:** Similar to R-CNN, the image is divided into regions using a selective search algorithm.
2. **Feature Extraction:** Instead of using a separate CNN for each region proposal, Fast R-CNN uses a single CNN to extract features from the entire image. The selective search regions are then warped to a fixed size and fed into the CNN as input.
3. **RoI Pooling:** Fast R-CNN uses a region of interest (RoI) pooling layer to extract a fixed-length feature vector from each region proposal. The RoI pooling layer divides each proposal into a fixed number of sub-windows and applies max-pooling to each sub-window, resulting in a fixed-size feature map that can be fed into a fully connected layer.
4. **Object Classification and Localization:** Fast R-CNN uses a single network to perform both object classification and localization. The output of the RoI pooling layer is fed into a series of fully connected layers that produce class probabilities and bounding box coordinates for each region proposal.

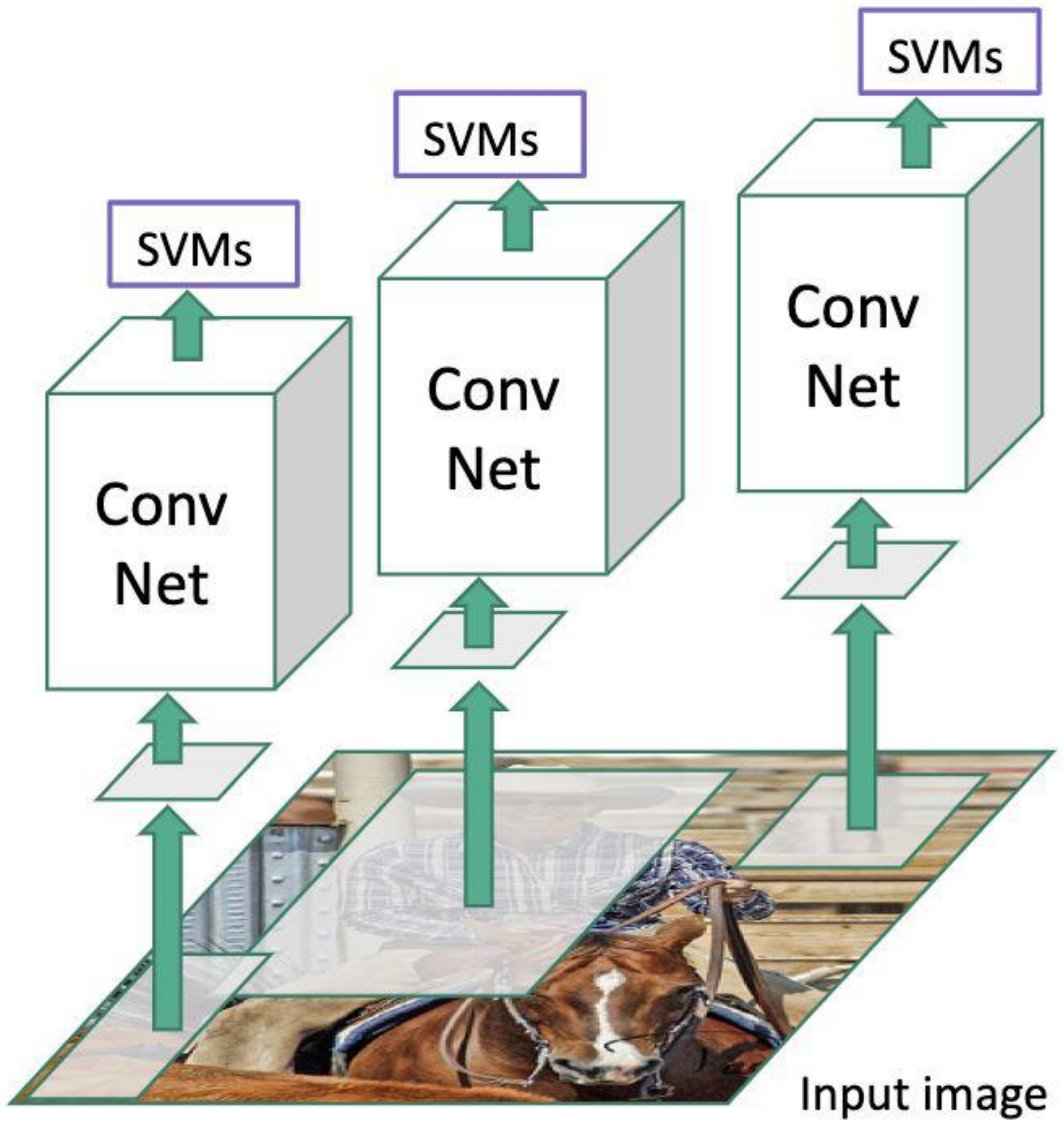
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Fast R-CNN



“Slow” R-CNN

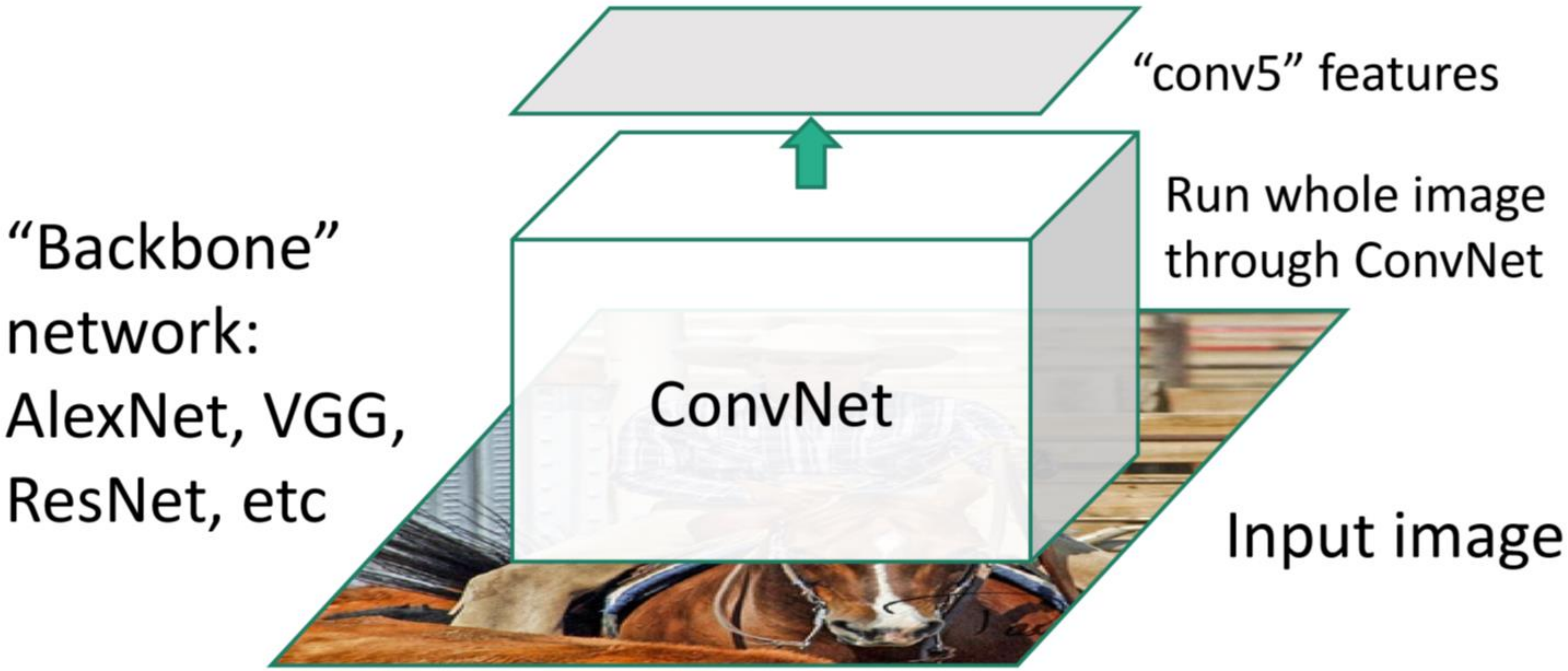


Girshick, “Fast R-CNN”, ICCV 2015.

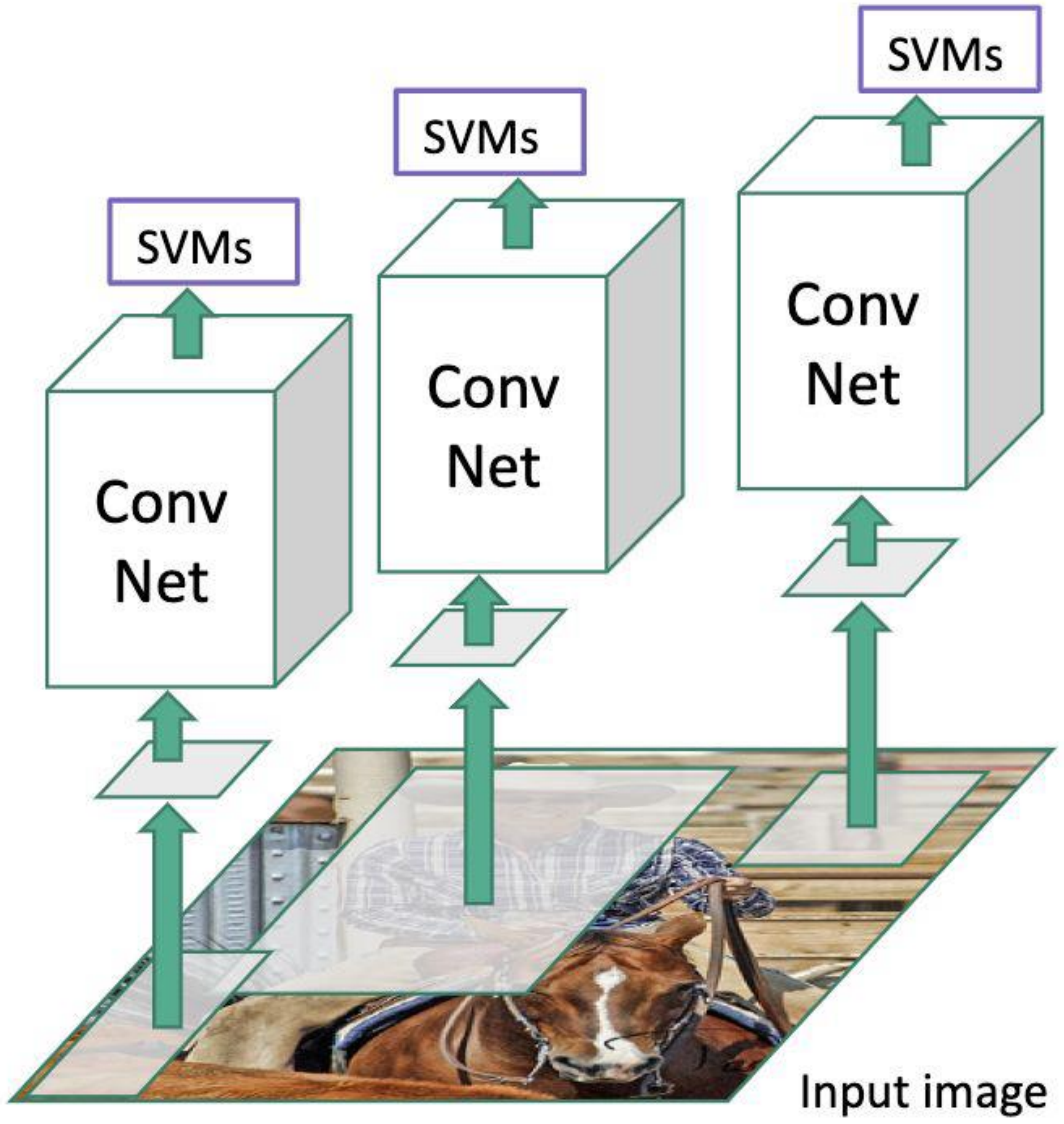
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Fast R-CNN



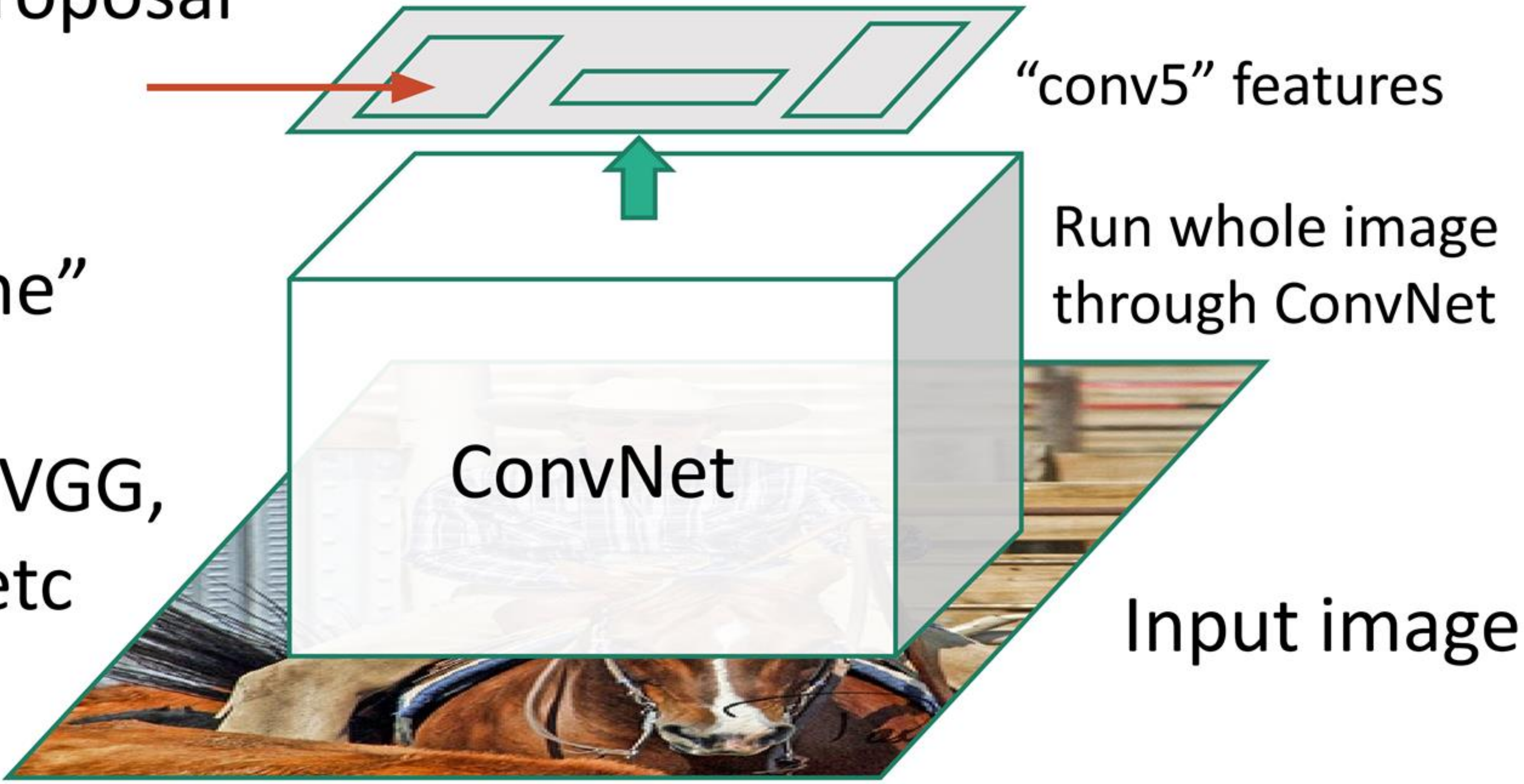
"Slow" R-CNN



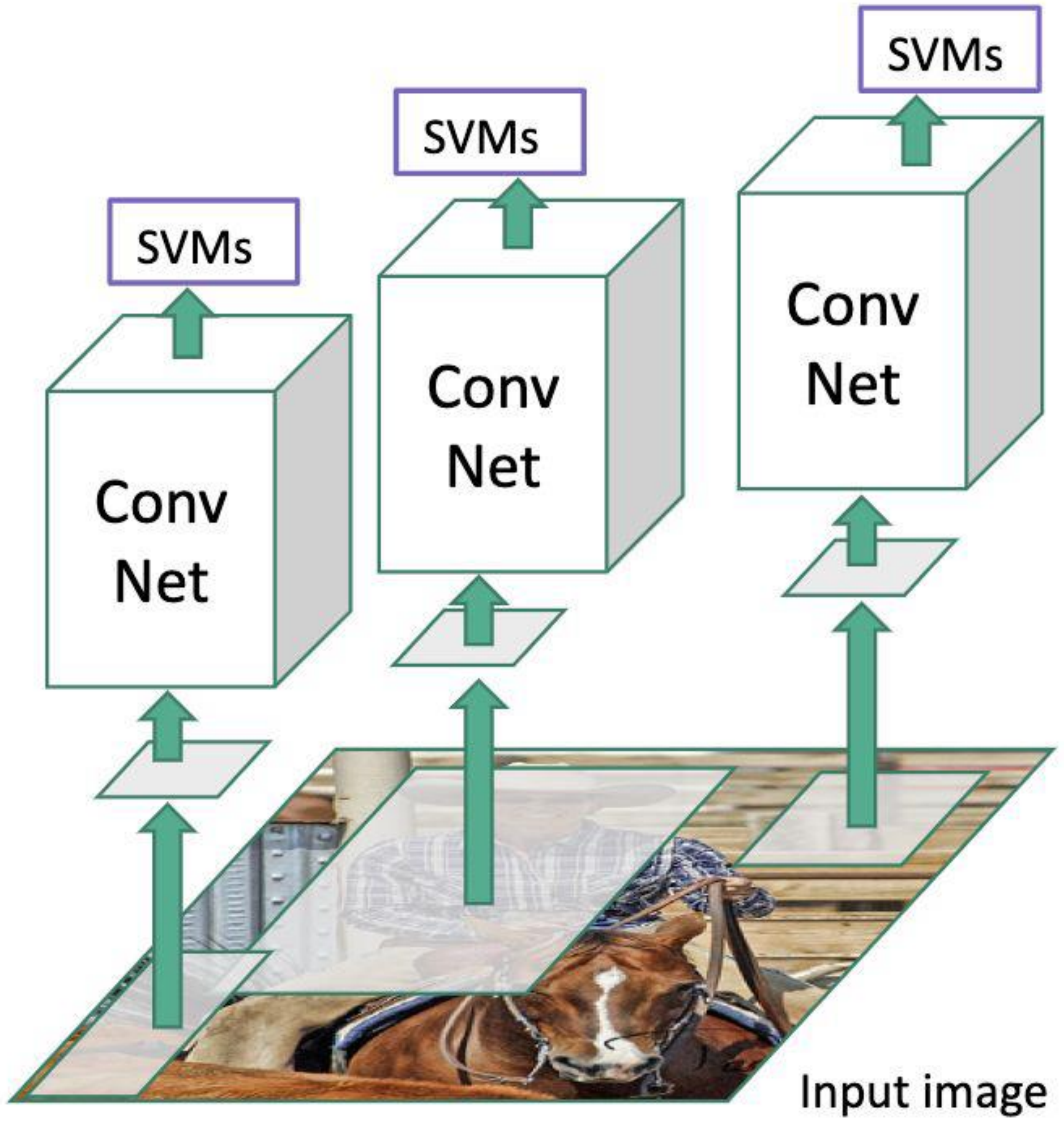
Girshick, "Fast R-CNN", ICCV 2015.

Regions of Interest (RoIs) from a proposal method

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



"Slow" R-CNN



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

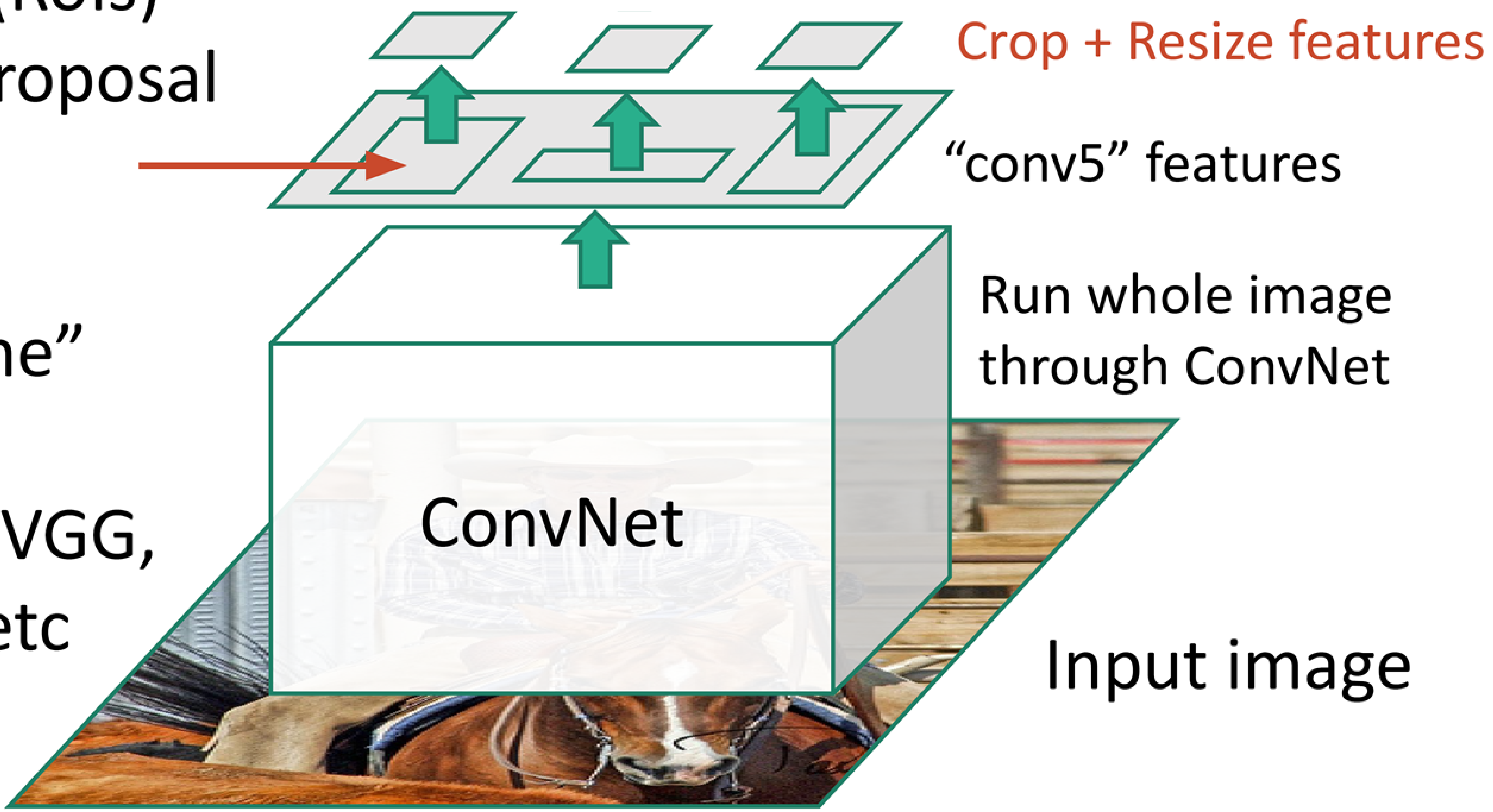
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

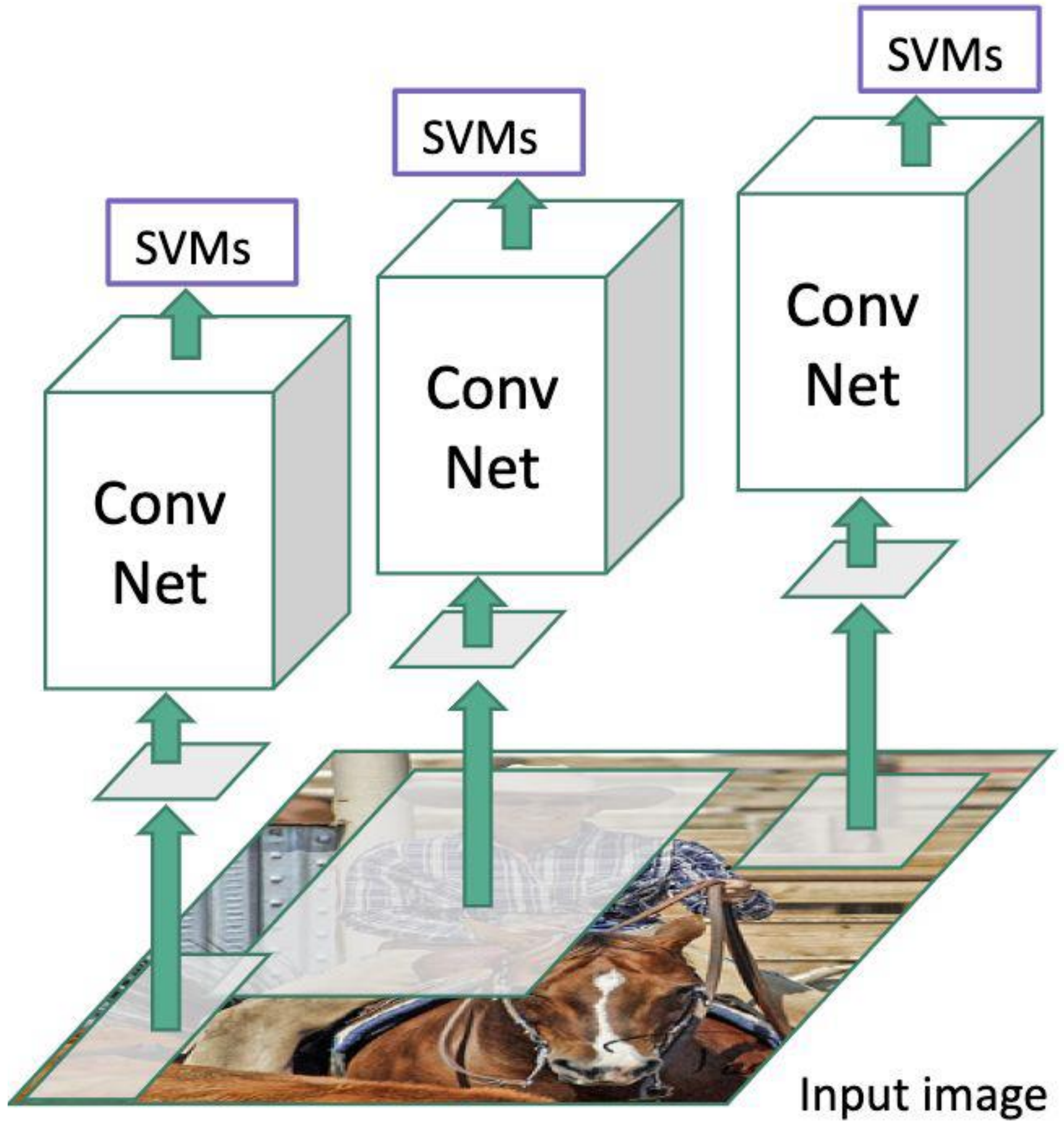
Fast R-CNN

Regions of Interest (RoIs) from a proposal method

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

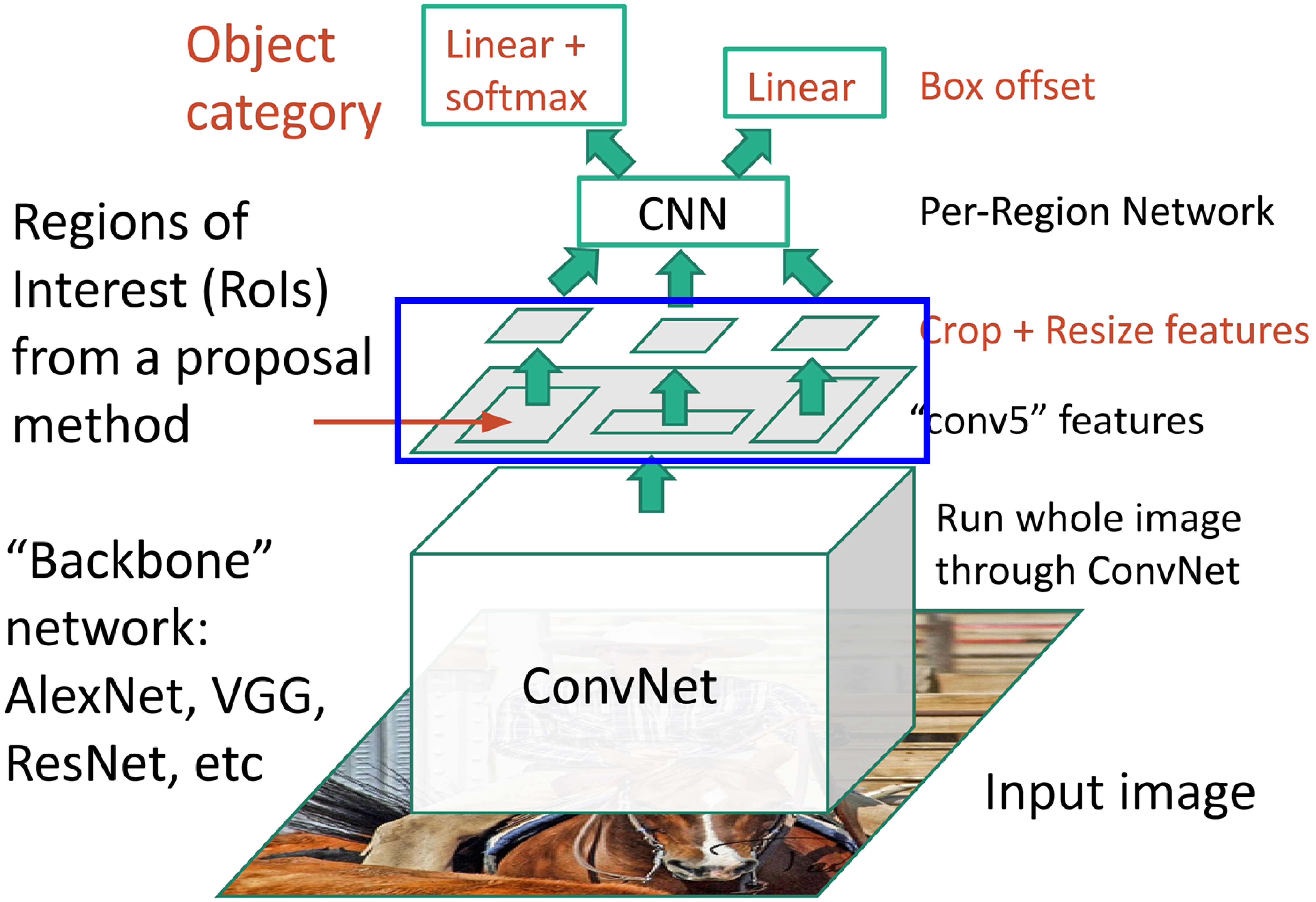


"Slow" R-CNN

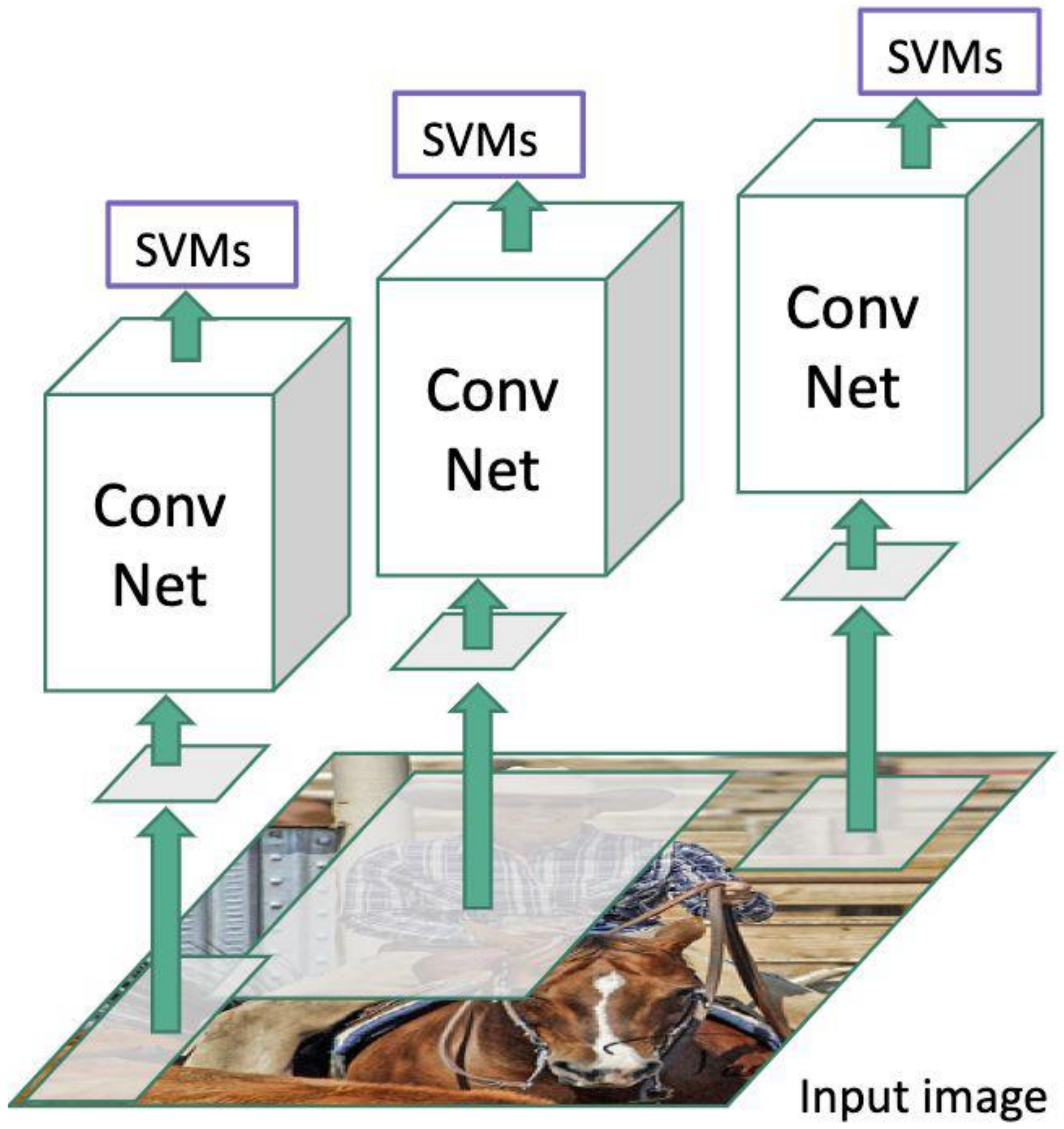


Girshick, "Fast R-CNN", ICCV 2015.

Fast R-CNN



"Slow" R-CNN



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

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Fast R-CNN: *Advantages over R-CNN*

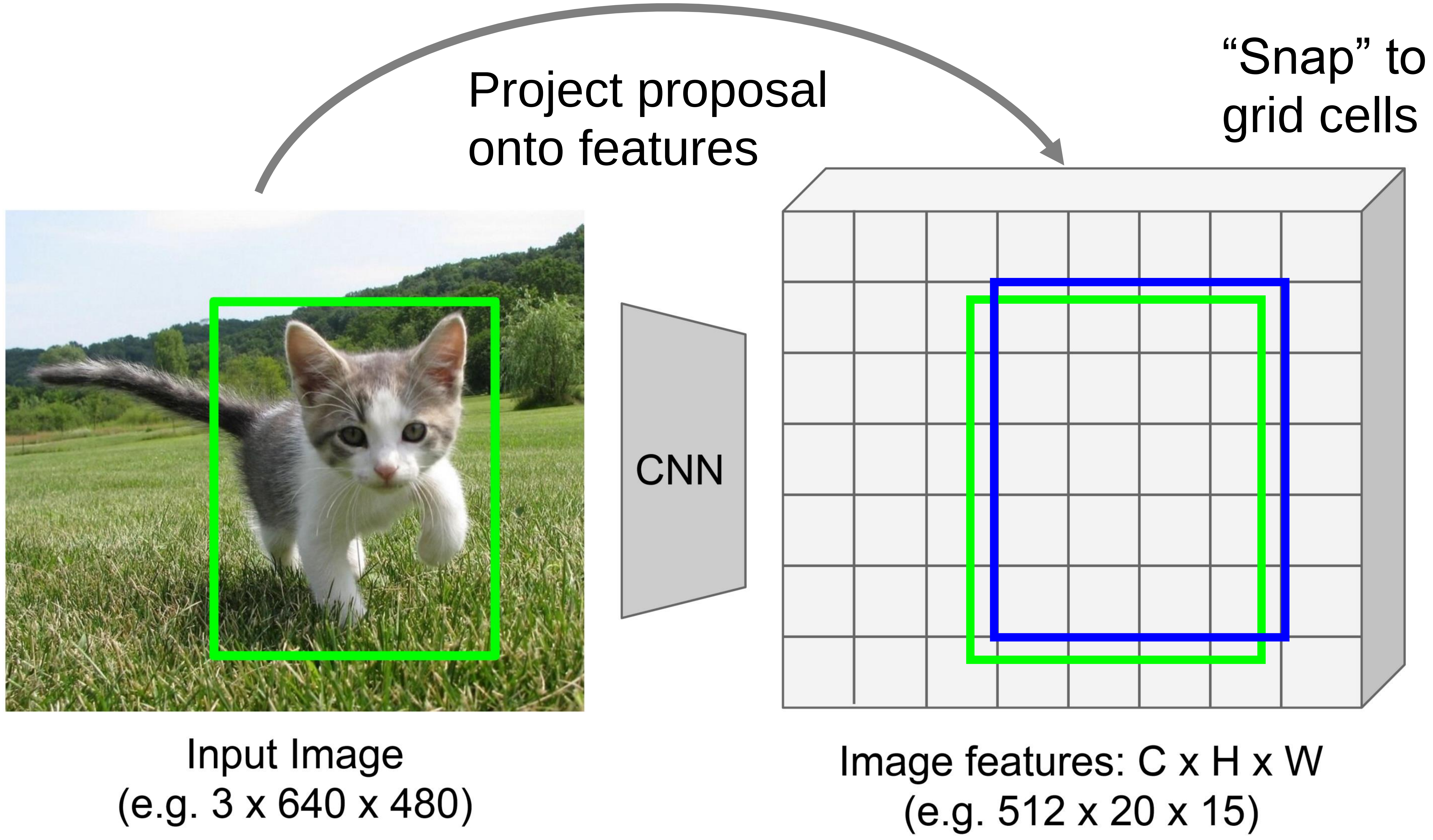
1. **Faster Training and Inference:** Since Fast R-CNN uses a single CNN to extract features from the entire image, it can process multiple region proposals in parallel, resulting in faster training and inference times compared to R-CNN.

2. **End-to-End Training:** Fast R-CNN enables end-to-end training of the entire system, which means that all components are optimized together to improve overall performance.

3. **Better Localization Accuracy:** Fast R-CNN uses the RoI pooling layer to extract features from each region proposal, resulting in more accurate object localization compared to R-CNN.

4. **Reduced Memory Usage:** Fast R-CNN requires less memory compared to R-CNN since it only needs to store the feature map for the entire image, instead of storing intermediate results for each region proposal.

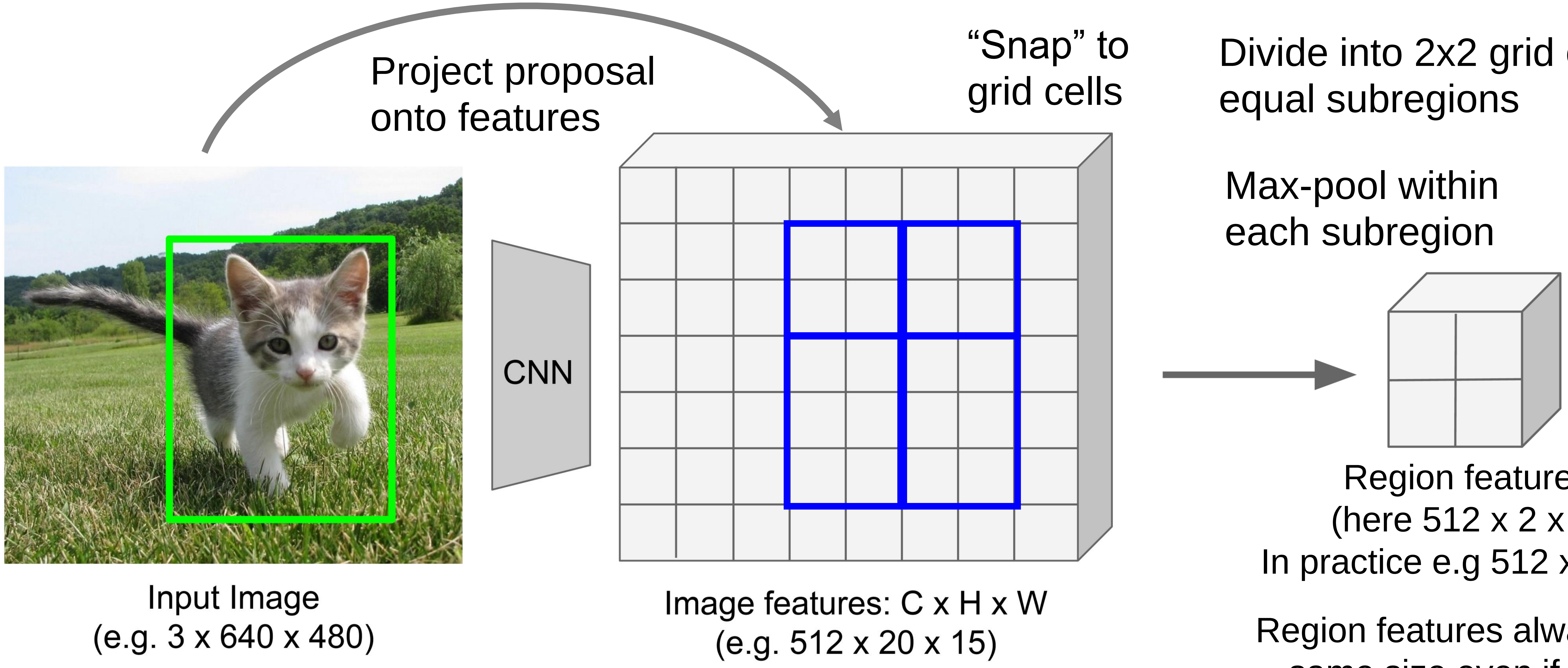
Cropping Features: *RoI Pool*



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

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

Cropping Features: *RoI Pool*



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!

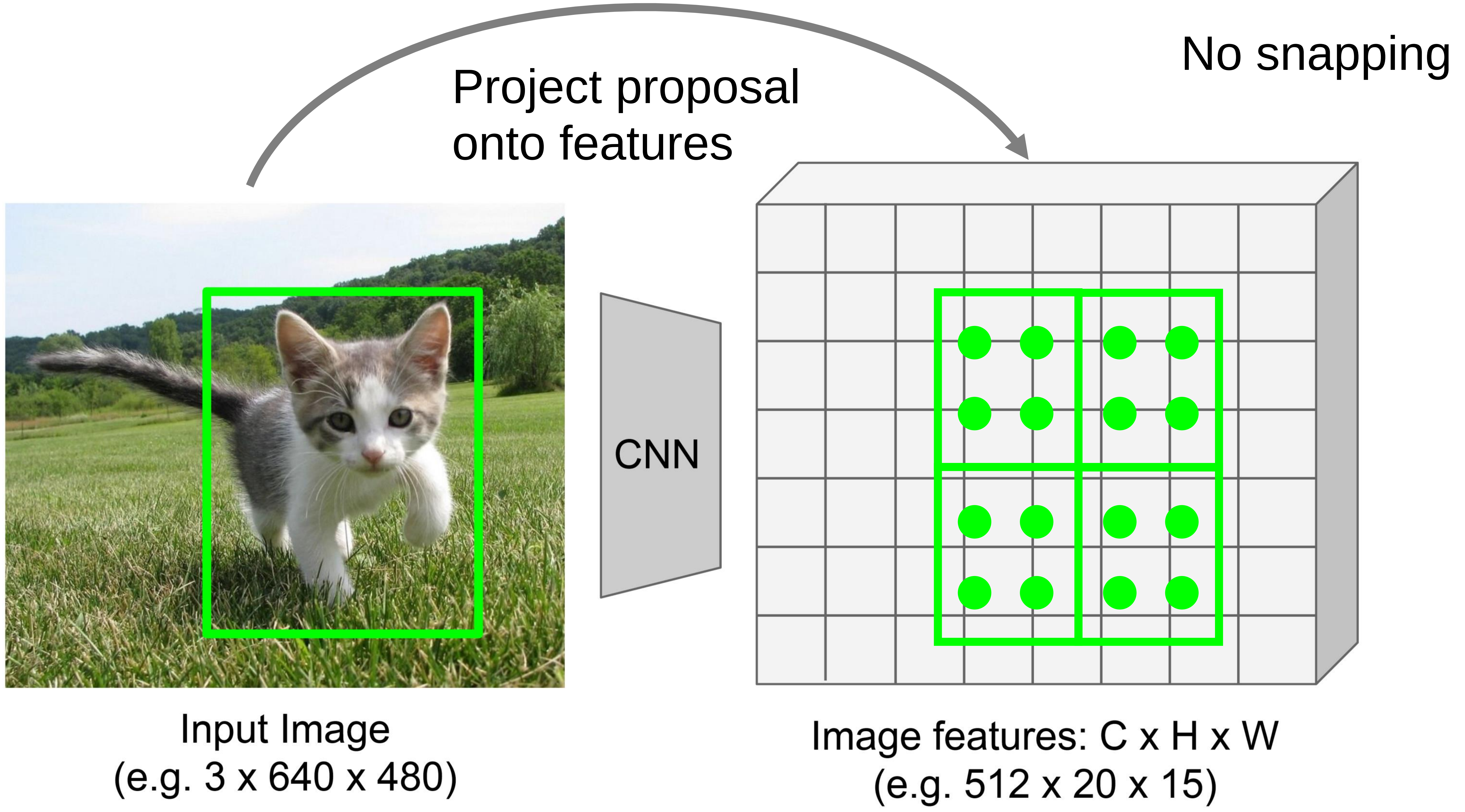
Girshick, "Fast R-CNN", ICCV 2015.

Problem: Region features slightly misaligned

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Cropping Features: *RoI Align*



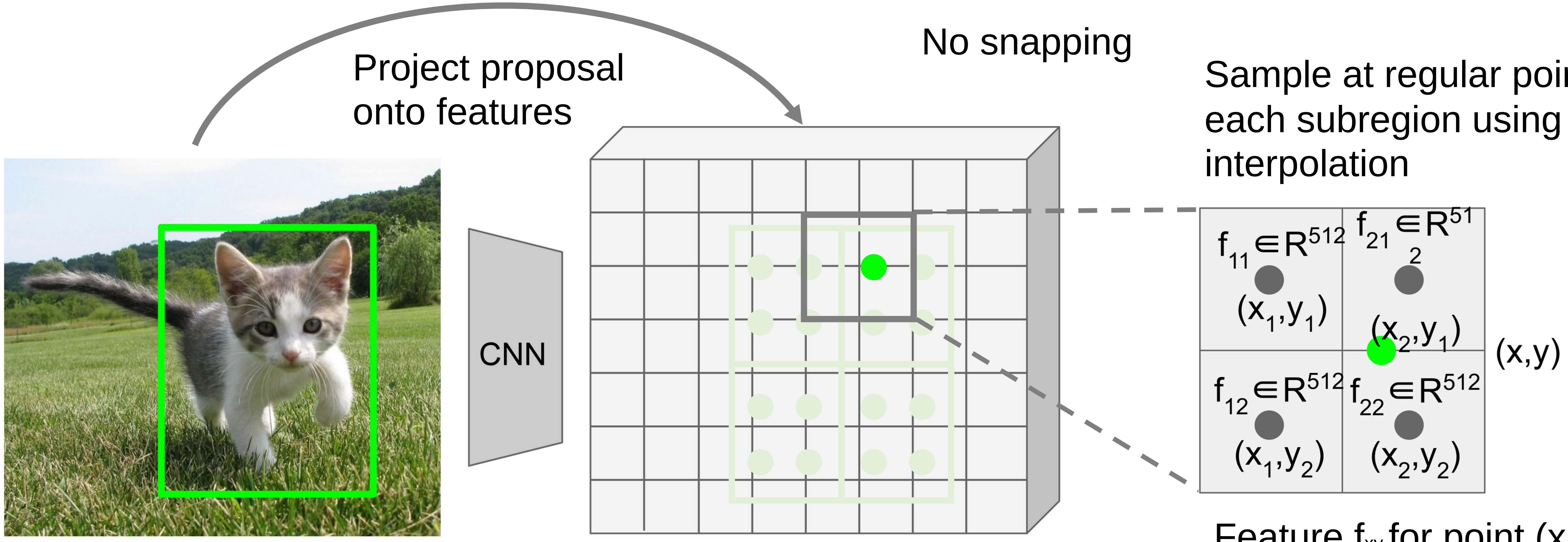
Sample at regular points in each subregion using bilinear interpolation

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

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Cropping Features: *RoI Align*



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

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

Sample at regular 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:

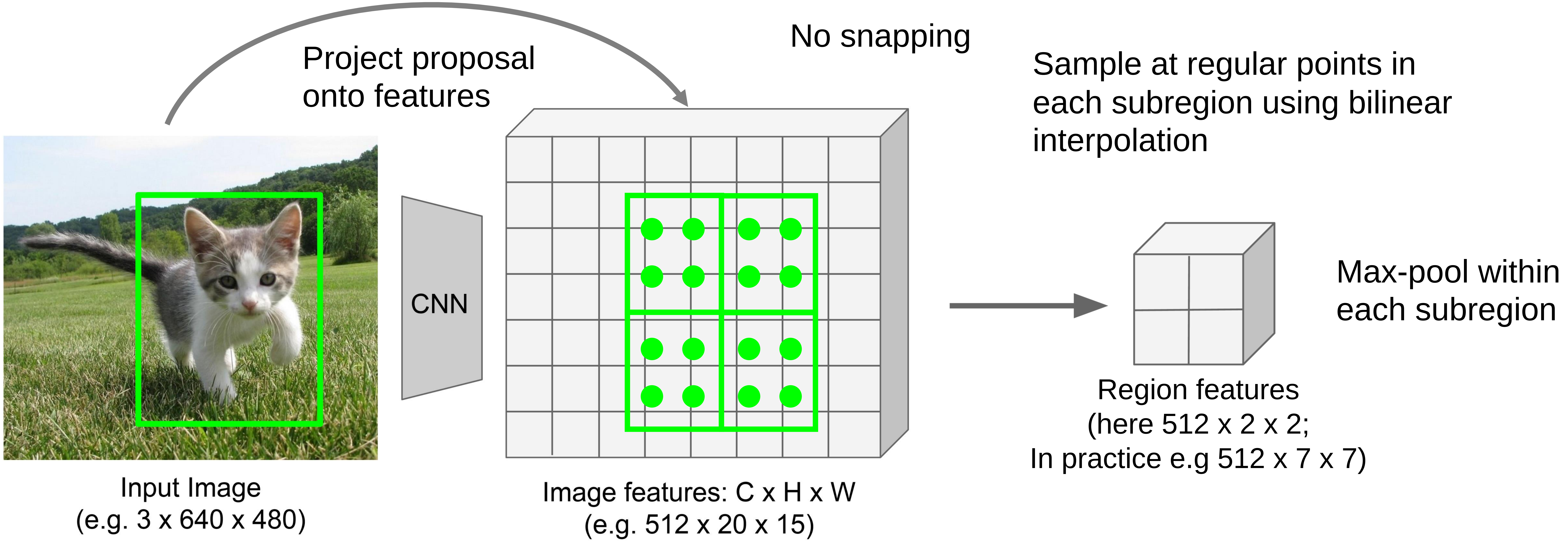
$$f_{xy} = \sum_{i,j=1}^{2} f_{i,j} \max(0, 1 - |x - x_i|) \max(0, 1 - |y - y_j|)$$

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

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Cropping Features: *RoI Pool*

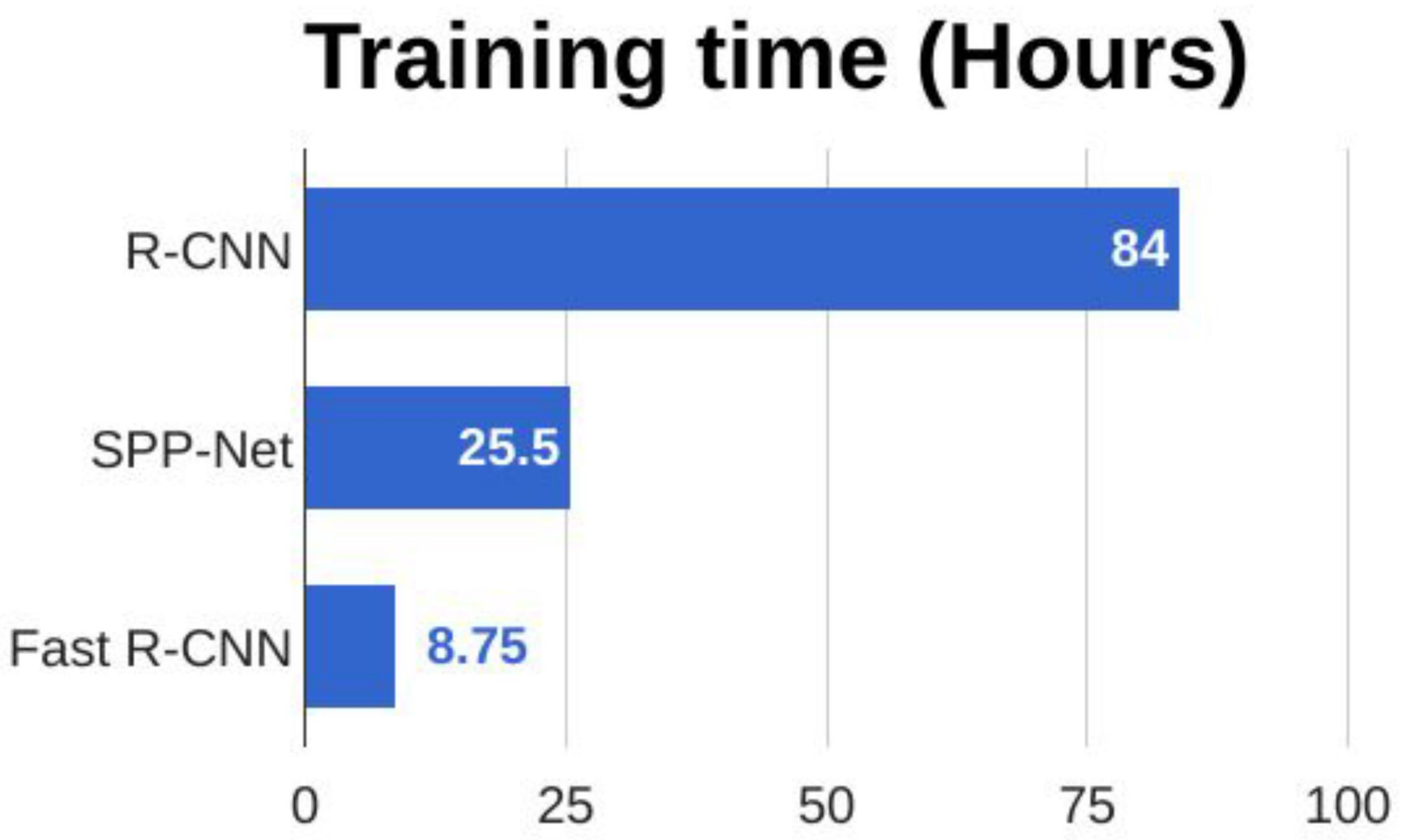


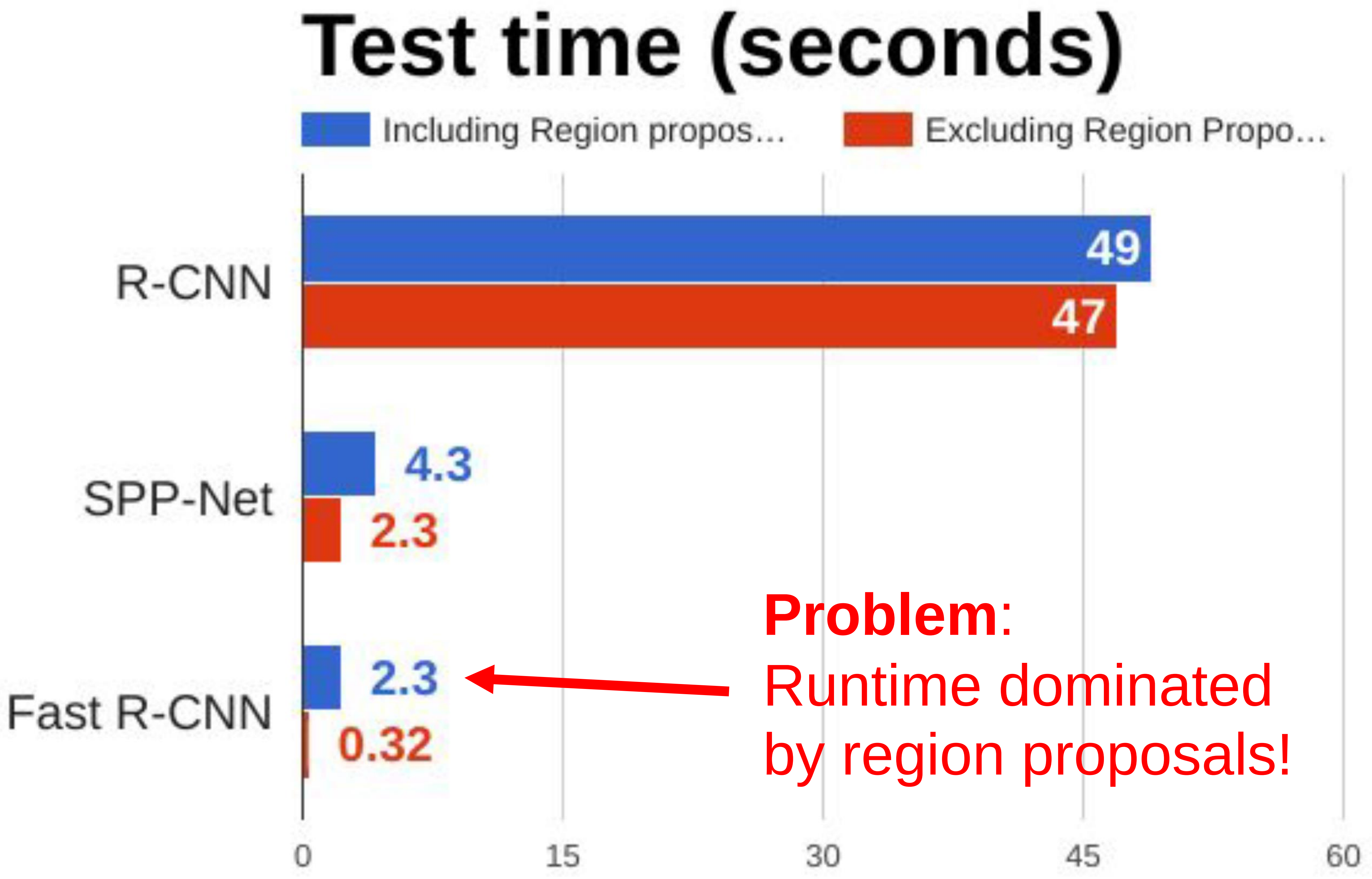
Girshick, “Fast R-CNN”, ICCV 2015.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Cropping Features: *RoI Pool*





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

Faster R-CNN: *Make CNN do proposals!*

Faster R-CNN is another improvement that addresses the limitations of previous networks by introducing a novel region proposal network (RPN). The RPN generates region proposals directly from the feature map, eliminating the need for the computationally expensive selective search algorithm used in R-CNN and Fast R-CNN.

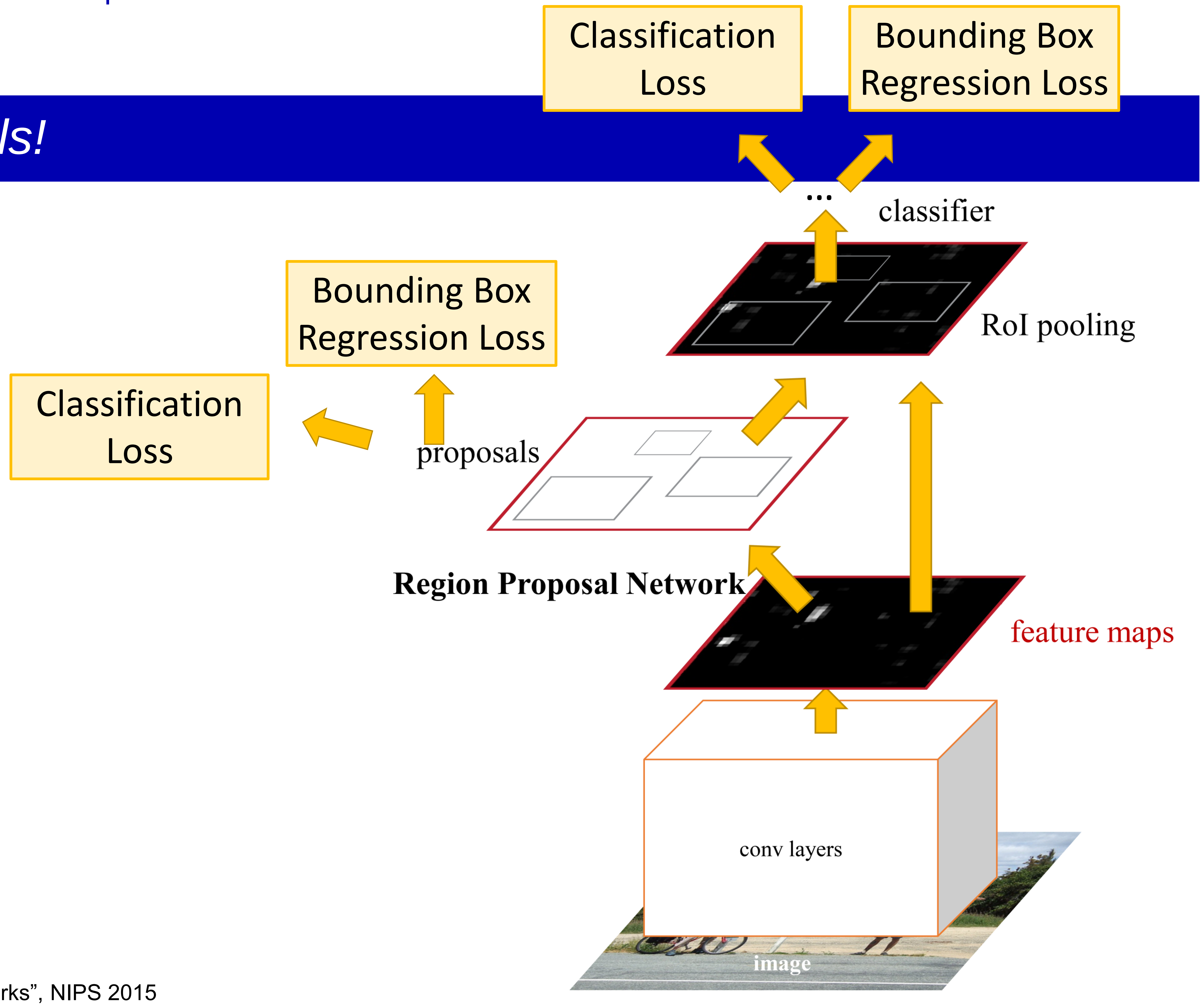
Here's how Faster R-CNN works:

1. **Feature Extraction:** Similar to Fast R-CNN, Faster R-CNN uses a single CNN to extract features from the entire image.
2. **Region Proposal Network:** The RPN is a small CNN that takes the feature map as input and outputs a set of object proposals along with their objectness scores. The proposals are generated by sliding a small network, known as an anchor box, over the feature map at different scales and aspect ratios.
3. **RoI Pooling:** The RoI pooling layer is used to extract a fixed-length feature vector from each region proposal, similar to Fast R-CNN.
4. **Object Classification and Localization:** The output of the RoI pooling layer is fed into a series of fully connected layers that produce class probabilities and bounding box coordinates for each region proposal.

Faster R-CNN: *Make CNN do 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

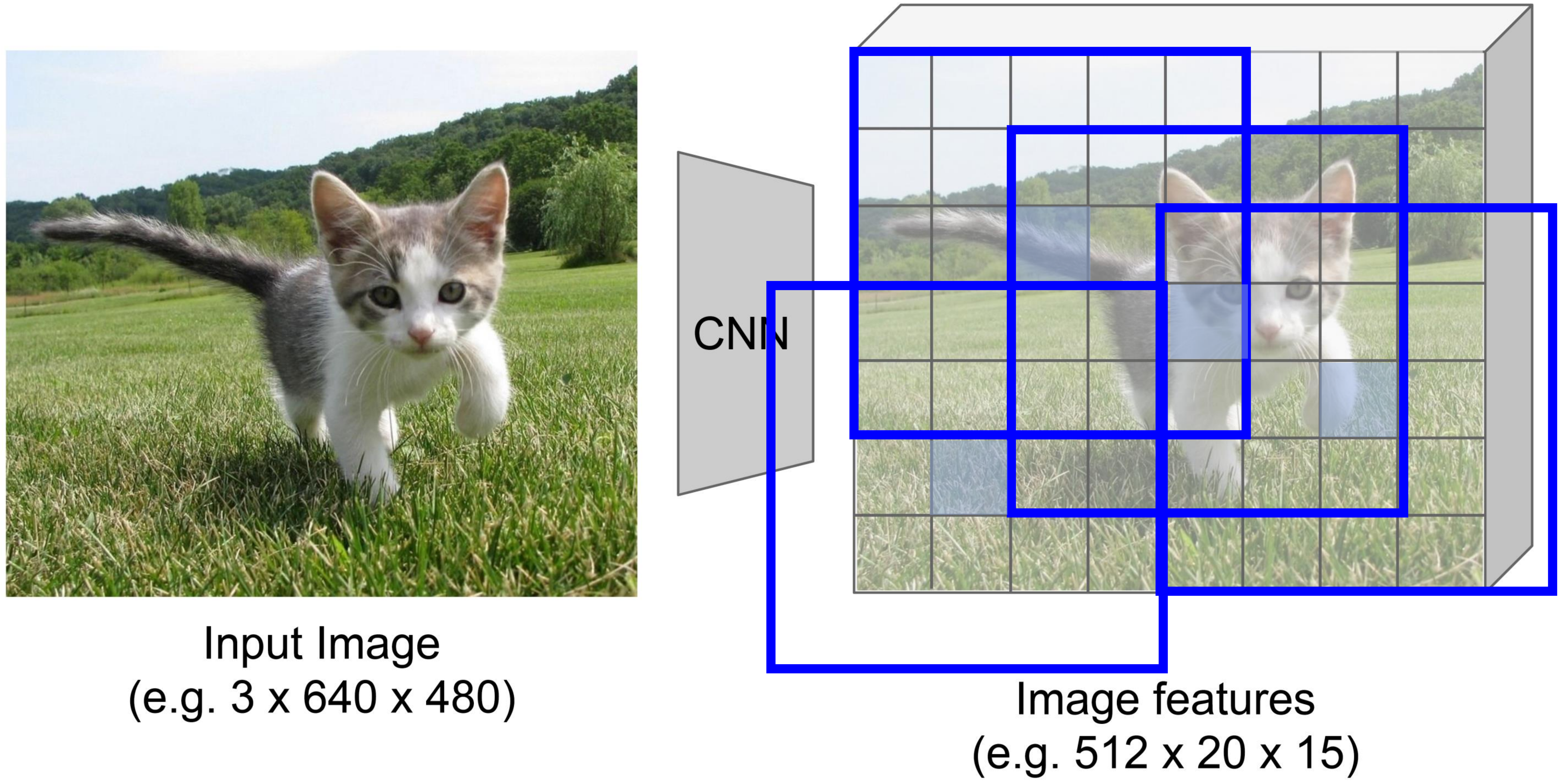


Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

Co-financed by the European Union
Connecting Europe Facility

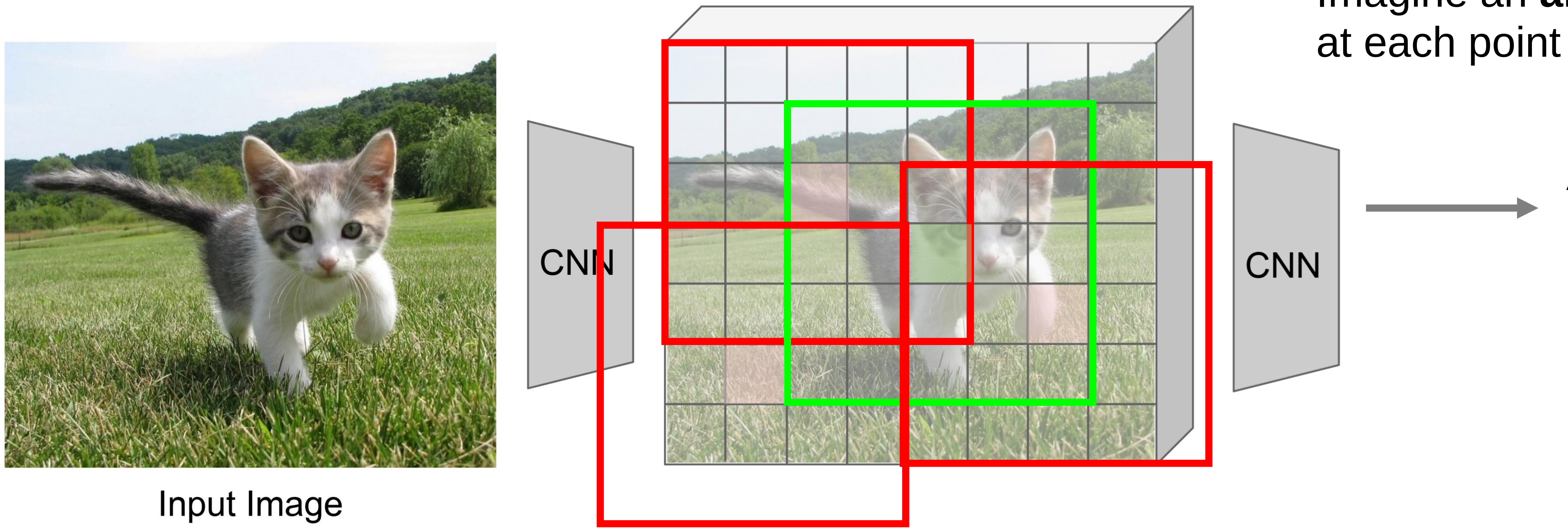
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Region Proposal Network



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

Region Proposal Network



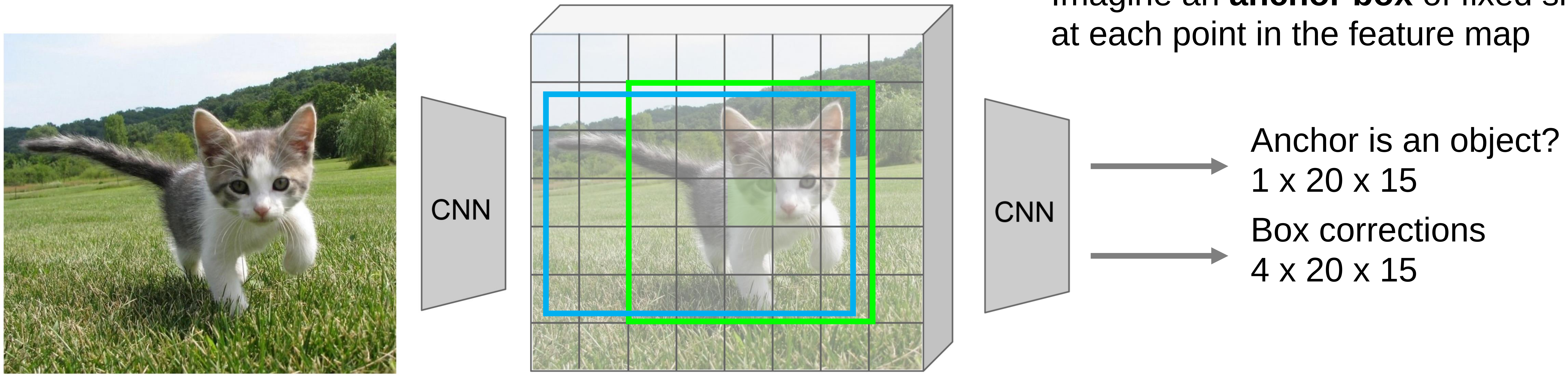
Input Image
(e.g. 3 x 640 x 480)

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

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

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

Region Proposal Network



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

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

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

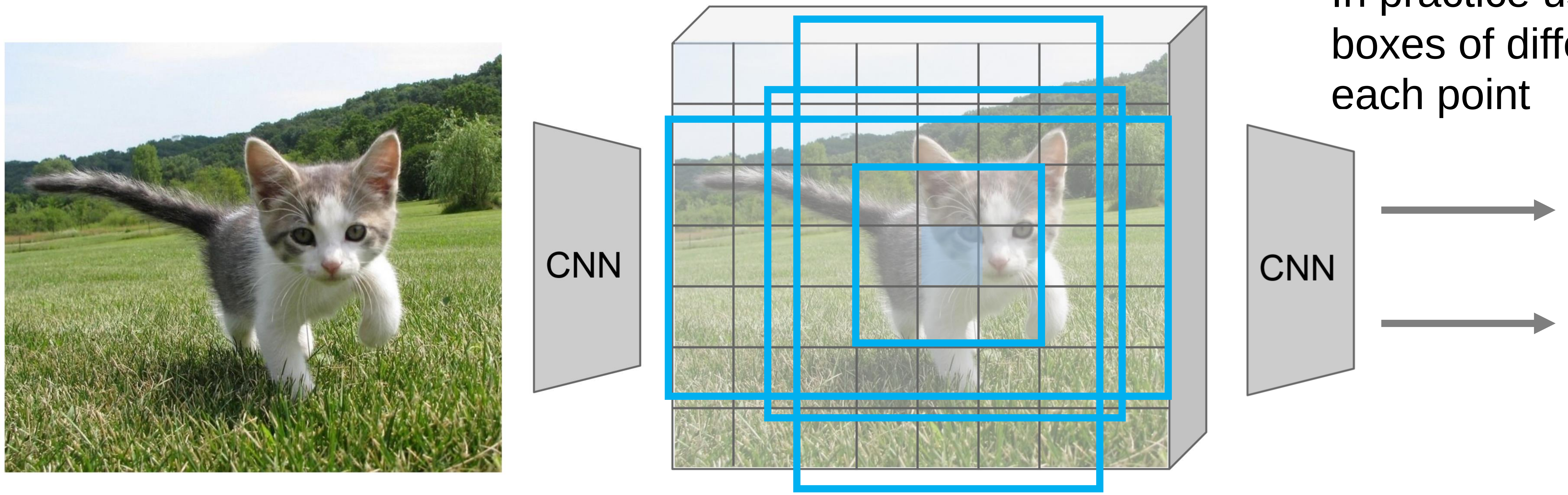
At each point, predict whether the corresponding anchor contains an object (binary classification)

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Region Proposal Network

In practice use K different anchor boxes of different size / scale at each point



Anchor is an object?
1 x 20 x 15

Box corrections
4 x 20 x 15

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

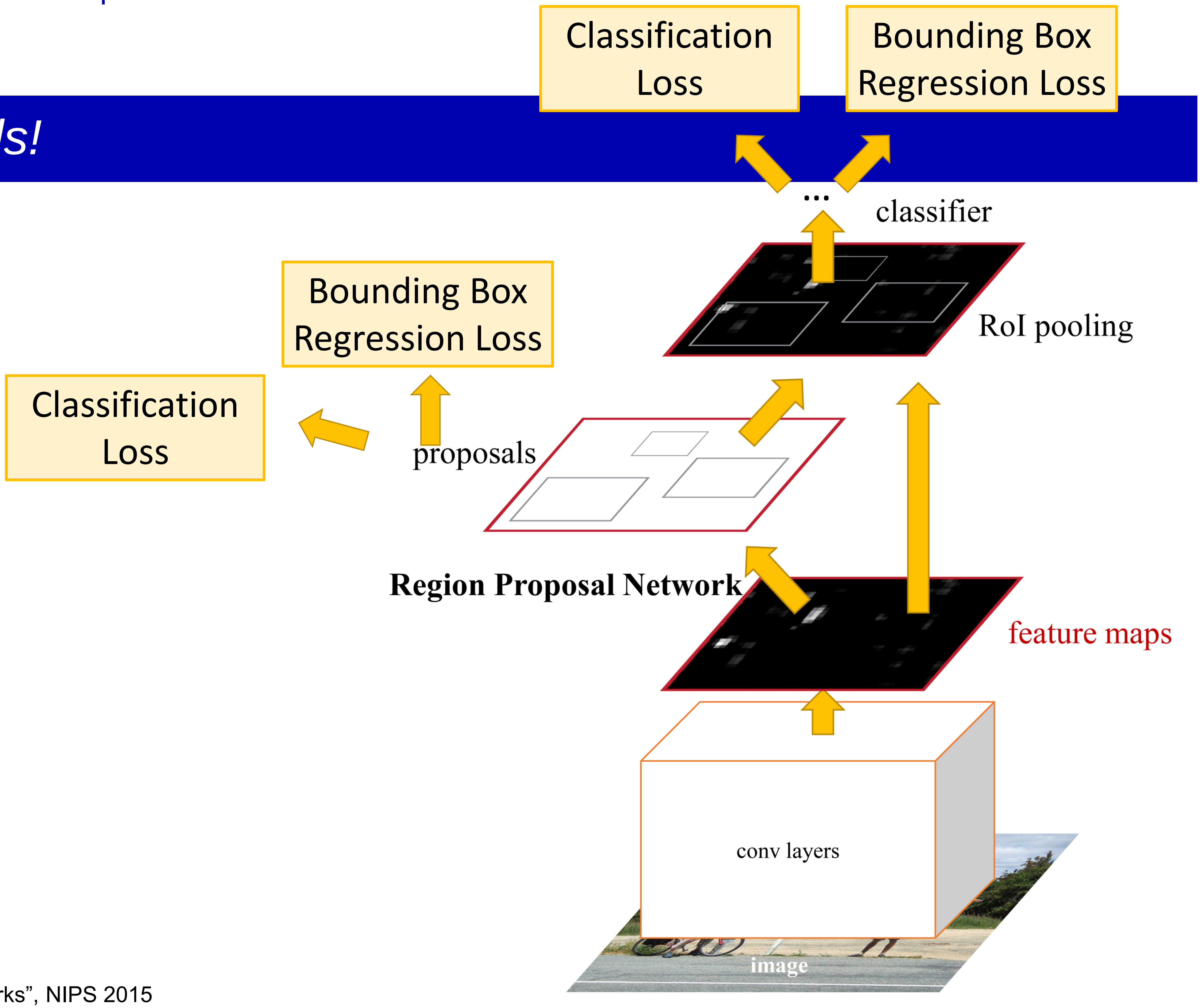
Image features
(e.g. 512 x 20 x 15)

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

Faster R-CNN: *Make CNN do proposals!*

Jointly train with 4 losses:

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

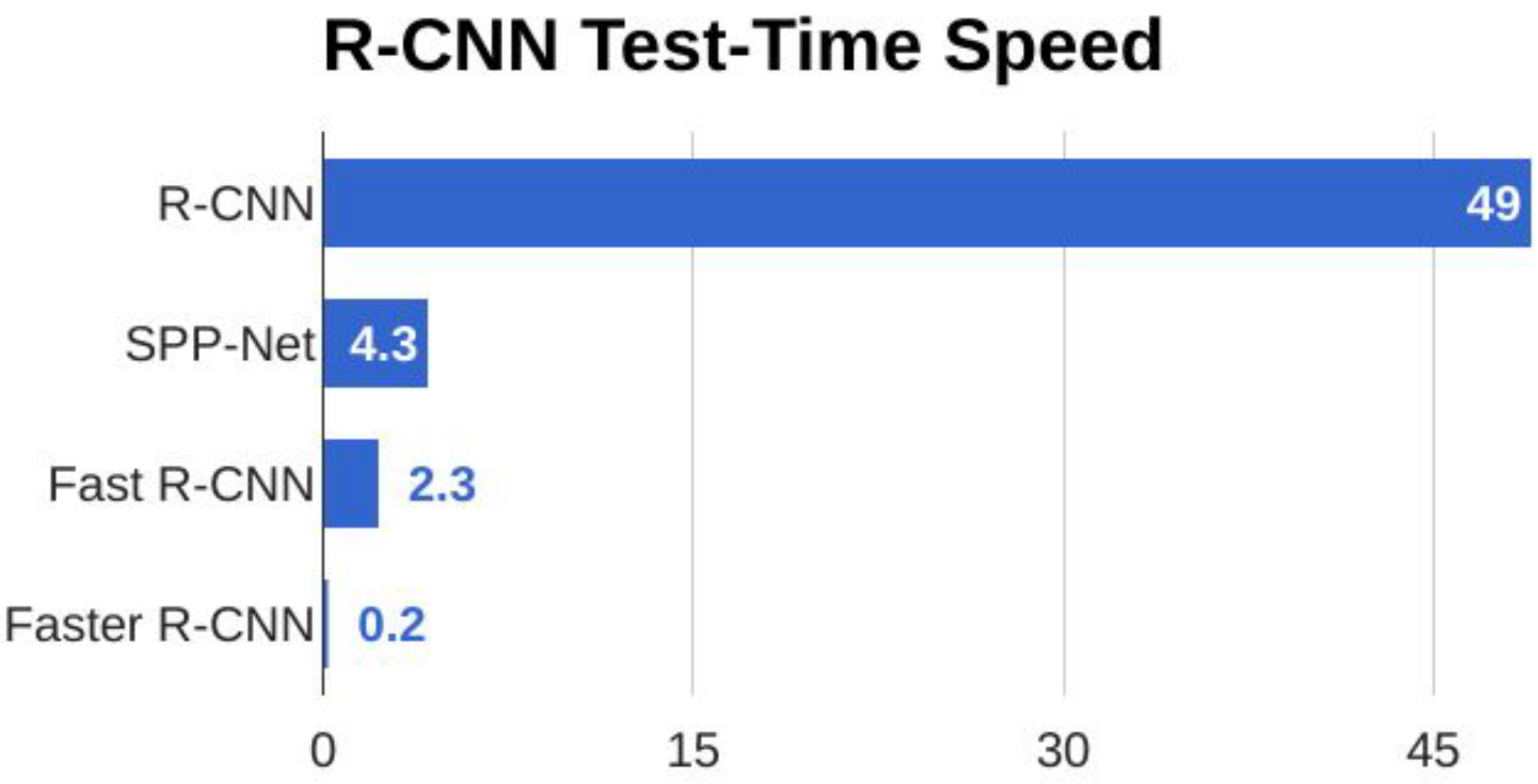


Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Faster R-CNN: *Make CNN do proposals!*



Faster R-CNN: *Advantages*

Faster Training and Inference: Faster R-CNN is faster than Fast R-CNN because it eliminates the need for the computationally expensive selective search algorithm used in Fast R-CNN. The RPN generates region proposals directly from the feature map, resulting in faster training and inference times.

Improved Localization Accuracy: Faster R-CNN is more accurate than Fast R-CNN because the RPN generates more accurate region proposals than the selective search algorithm used in Fast R-CNN.

Flexibility: The RPN in Faster R-CNN can be trained end-to-end with the rest of the network, enabling the system to learn more discriminative features for region proposal generation.

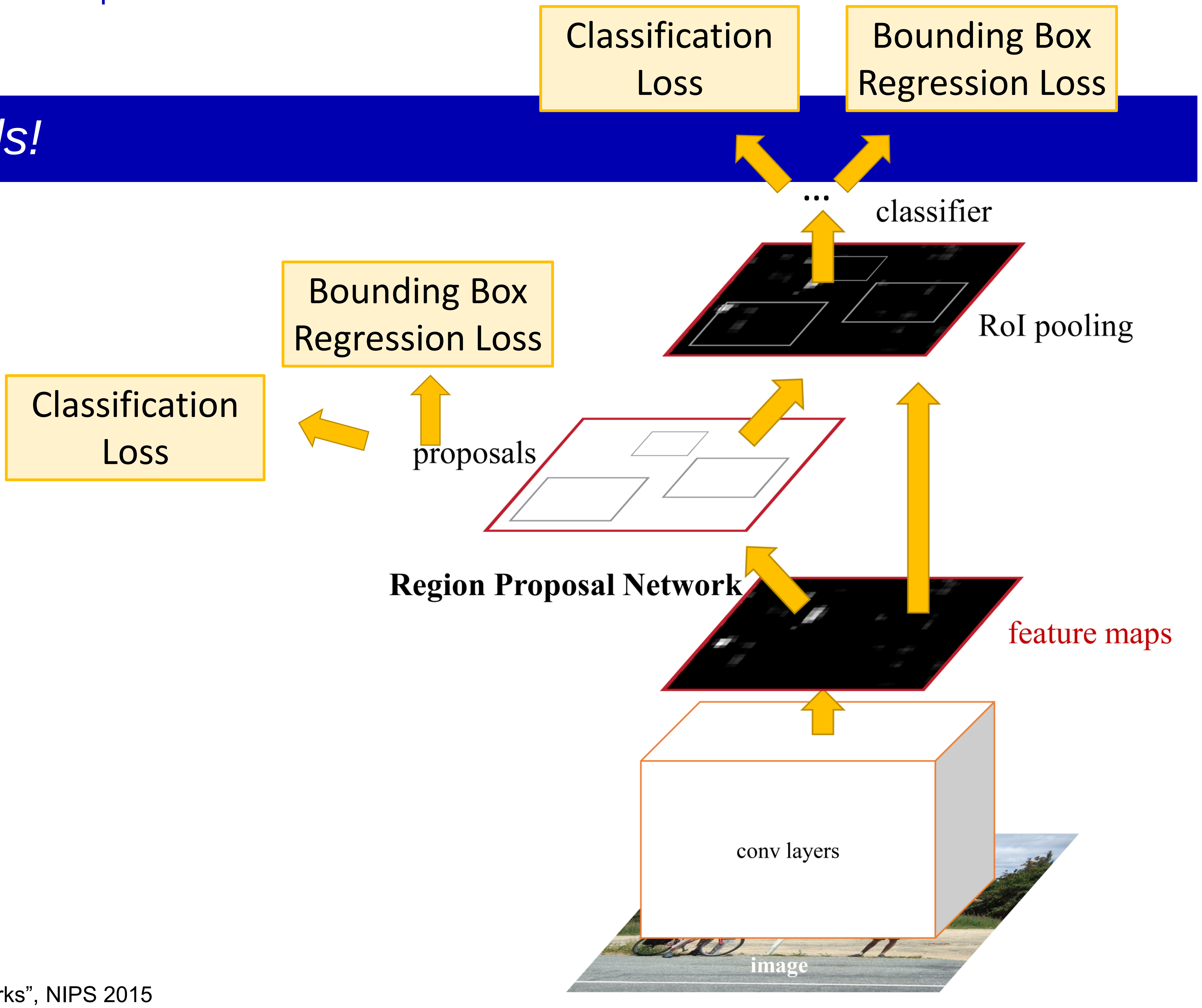
Adaptability: The RPN can be modified to handle various input sizes and aspect ratios, making it adaptable to a wide range of object detection tasks.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Faster R-CNN: *Make CNN do proposals!*

Glossing over many details:

- Ignore overlapping proposals with non-max suppression
- How are anchors determined?
- How do we sample positive / negative samples for training the RPN?
- How to parameterize bounding box regression?



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

Faster R-CNN: *Disadvantages*

Complexity: Faster R-CNN is a complex framework that requires a large number of layers and parameters, making it difficult to train and optimize. This can also result in high computational and memory requirements.

Anchors: The RPN in Faster R-CNN relies on anchor boxes to generate region proposals, which can be a limiting factor for certain types of objects and scenes. Choosing the right set of anchor sizes and aspect ratios is also challenging.

Object Scale: Faster R-CNN is designed to handle objects at a wide range of scales, but it may still struggle with objects that are too small or too large. The RPN may generate too many or too few region proposals for such objects, affecting detection accuracy.

Training Data: Like other deep learning models, Faster R-CNN requires large amounts of training data to achieve good performance. Collecting and annotating such data can be time-consuming and expensive.

Performance Tradeoffs: Like all object detection frameworks, Faster R-CNN involves a tradeoff between detection accuracy and inference speed. Higher accuracy usually requires more computation, while faster inference can result in lower accuracy.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

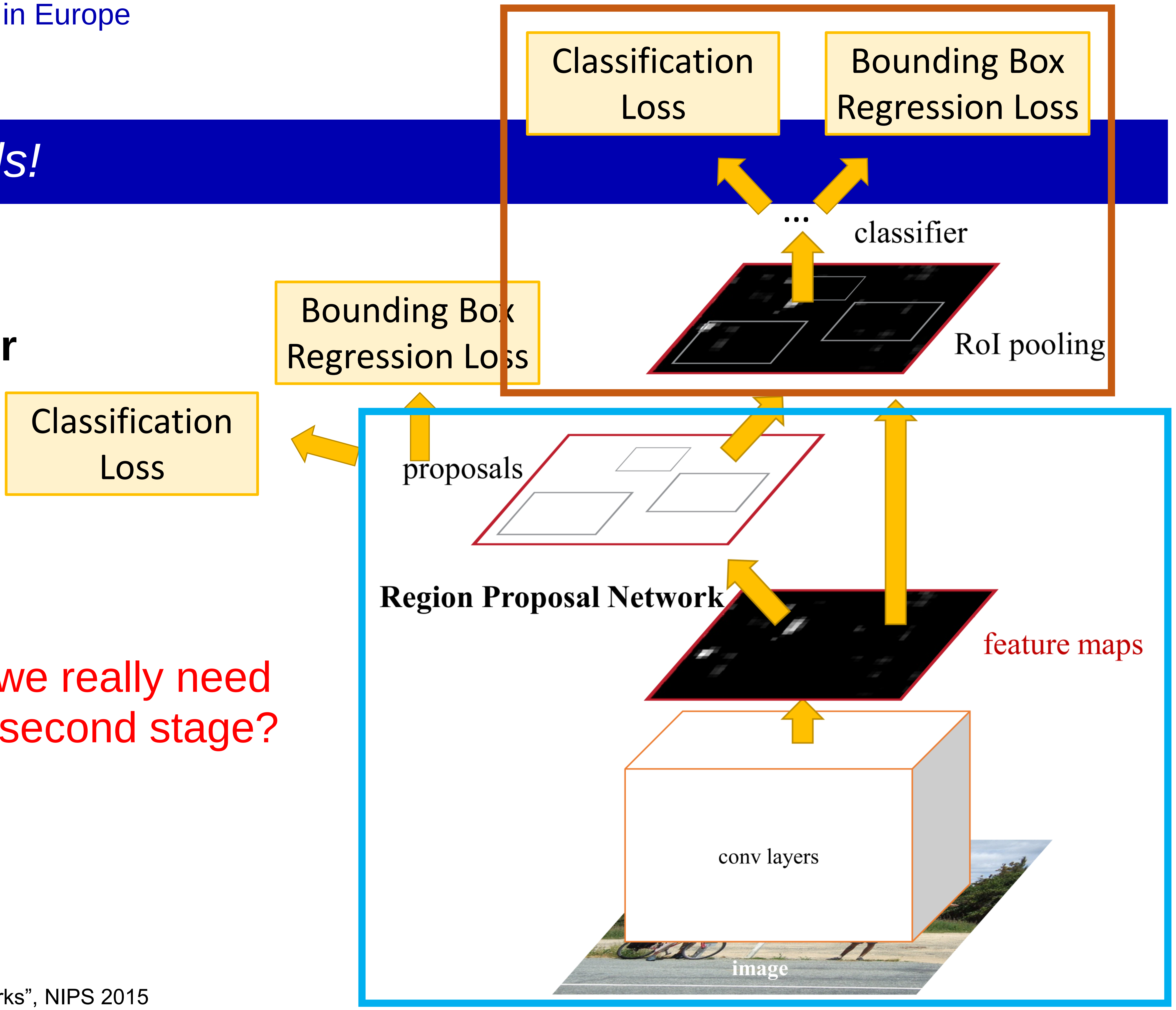
Faster R-CNN: *Make CNN do proposals!*

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

Do we really need the second stage?



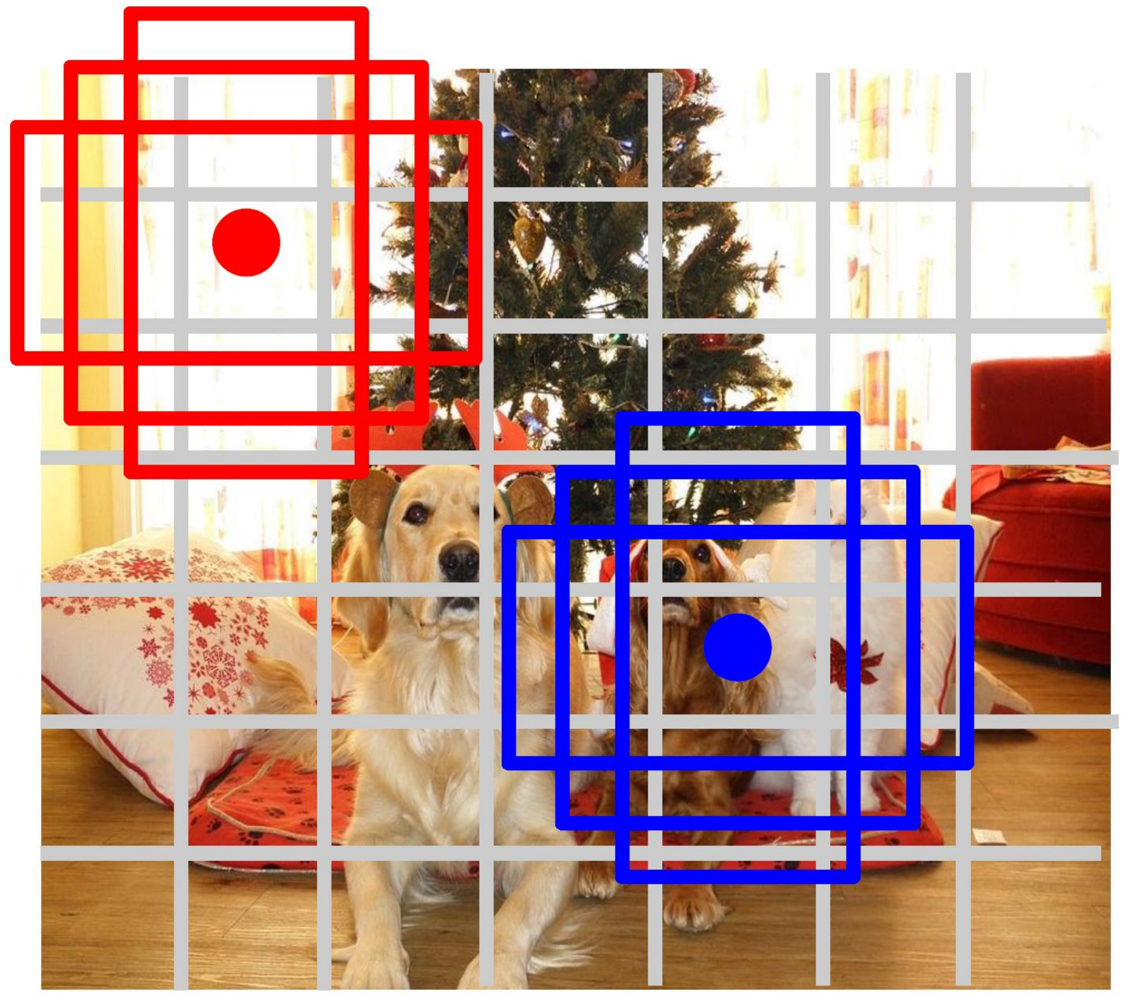
Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Single-Stage Object Detectors: *YOLO / SSD / RetinaNet*

Input image
3 x H x W



Divide image into grid
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
Liu et al, "SSD: Single-Shot MultiBox Detector", ECCV 2016
Lin et al, "Focal Loss for Dense Object Detection", ICCV 2017

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Single-Stage Object Detectors: *YOLO / SSD / RetinaNet*

YOLO (You Only Look Once) is a real-time object detection system that uses a single neural network to predict bounding boxes and class probabilities directly from full images in one evaluation, rather than using a sliding window approach or region proposal method.

The YOLO algorithm takes an input image and divides it into a grid of cells. For each cell, YOLO predicts the bounding boxes of objects that may be present in that cell, along with their associated class probabilities. The predictions are made using a single neural network that takes the entire image as input and outputs the predicted bounding boxes and class probabilities for all cells in a single pass.

YOLO is known for its speed and real-time performance, making it popular in applications such as self-driving cars, surveillance, and robotics. However, the tradeoff for its speed is lower accuracy compared to some other object detection algorithms.

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

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Single-Stage Object Detectors: *YOLO / SSD / RetinaNet*

SSD (Single-Shot MultiBox Detector) is a popular object detection algorithm that builds upon the previous work of the MultiBox object detection framework, but with improvements in terms of accuracy and speed.

Like YOLO, SSD is a real-time object detection algorithm that uses a single neural network to predict the bounding boxes and class probabilities for objects in an image. However, SSD differs from YOLO in that it uses multiple layers with different resolutions to detect objects of various sizes.

SSD generates a set of default bounding boxes of different aspect ratios and scales for each location in the feature maps, and then predicts the offset and class probabilities for these bounding boxes. By doing this, SSD can detect objects of various sizes and aspect ratios with high accuracy.

In general, SSD is known for its high accuracy and efficiency, making it a popular choice for real-time object detection applications such as robotics, self-driving cars, and surveillance systems.

Liu et al, “SSD: Single-Shot MultiBox Detector”, ECCV 2016

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Single-Stage Object Detectors: *YOLO / SSD / RetinaNet*

RetinaNet solves the problem of object detection in the presence of a large number of background regions, where most object detectors struggle due to the class imbalance problem.

RetinaNet uses a feature pyramid network (FPN) architecture to generate a set of feature maps with different resolutions. It then applies a novel focal loss function that down-weights the loss assigned to well-classified examples, making the training more robust to the class imbalance problem.

RetinaNet also uses a novel "one-stage" detection approach that combines the benefits of both "one-stage" and "two-stage" detection methods. Specifically, RetinaNet predicts object classification and bounding box regression in a single stage, similar to one-stage detectors like YOLO and SSD. However, it also uses a feature pyramid network and multiple levels of feature maps, similar to two-stage detectors like Faster R-CNN, to improve detection accuracy.

Lin et al, “Focal Loss for Dense Object Detection”, ICCV 2017

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Object Detection: *Takeaways*

Faster R-CNN is slower but more accurate

SSD is much faster but not as accurate

Bigger / Deeper backbones work better

Huang et al, “Speed/accuracy trade-offs for modern convolutional object detectors”, CVPR 2017
Zou et al, “Object Detection in 20 Years: A Survey”, arXiv 2019

R-FCN: Dai et al, “R-FCN: Object Detection via Region-based Fully Convolutional Networks”, NIPS 2016
Inception-V2: Ioffe and Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, ICML 2015
Inception V3: Szegedy et al, “Rethinking the Inception Architecture for Computer Vision”, arXiv 2016
Inception ResNet: Szegedy et al, “Inception-V4, Inception-ResNet and the Impact of Residual Connections on Learning”, arXiv 2016
MobileNet: Howard et al, “Efficient Convolutional Neural Networks for Mobile Vision Applications”, arXiv 2017

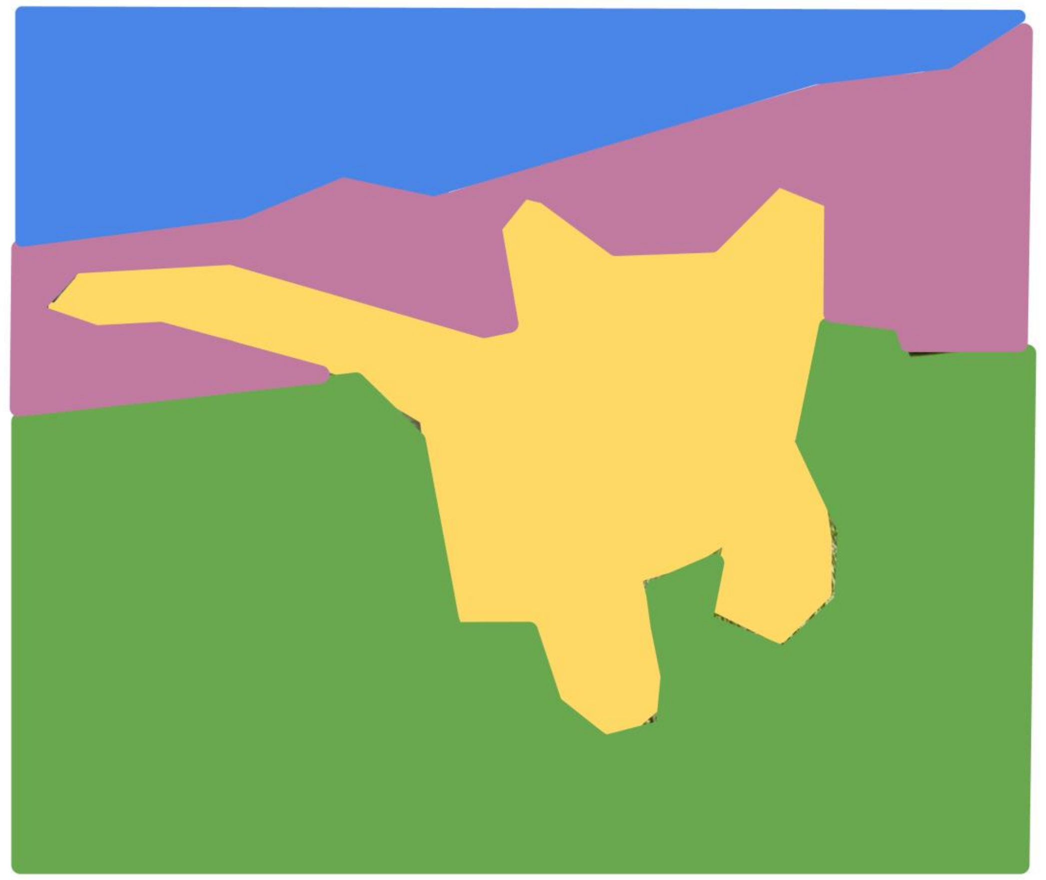
Instance Segmentation

Classification

CAT

No spatial extent

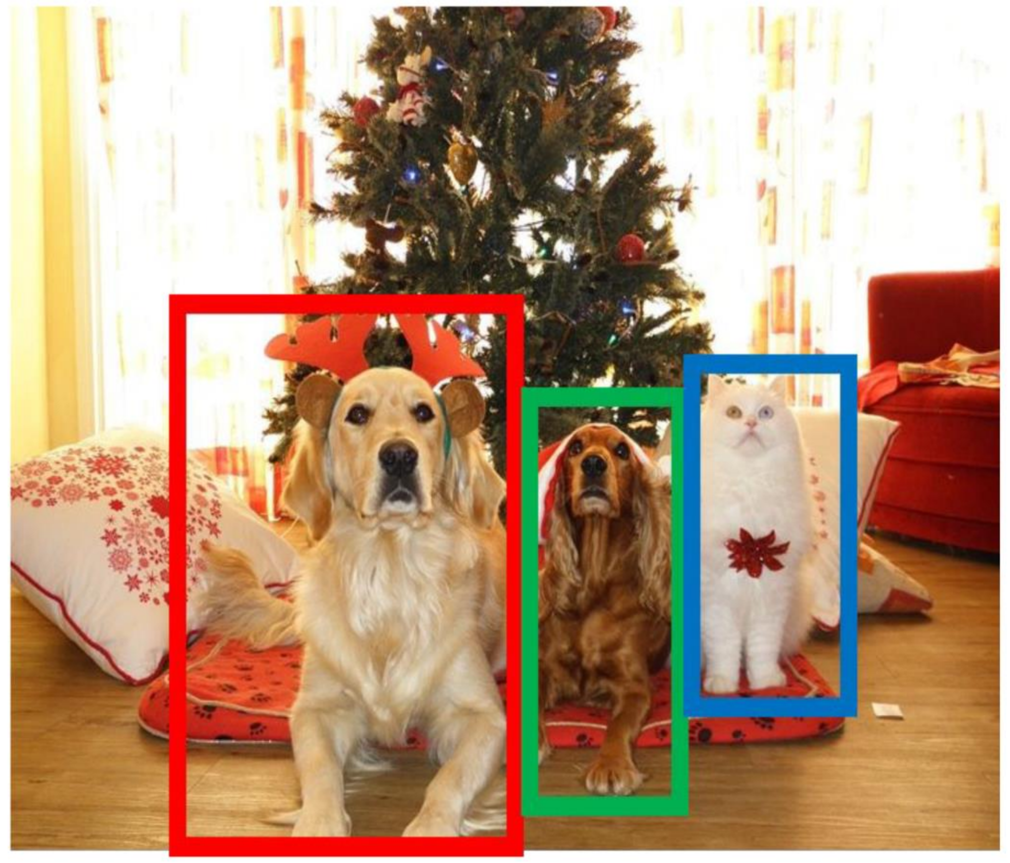
Semantic Segmentation



GRASS, CAT, TREE, SKY

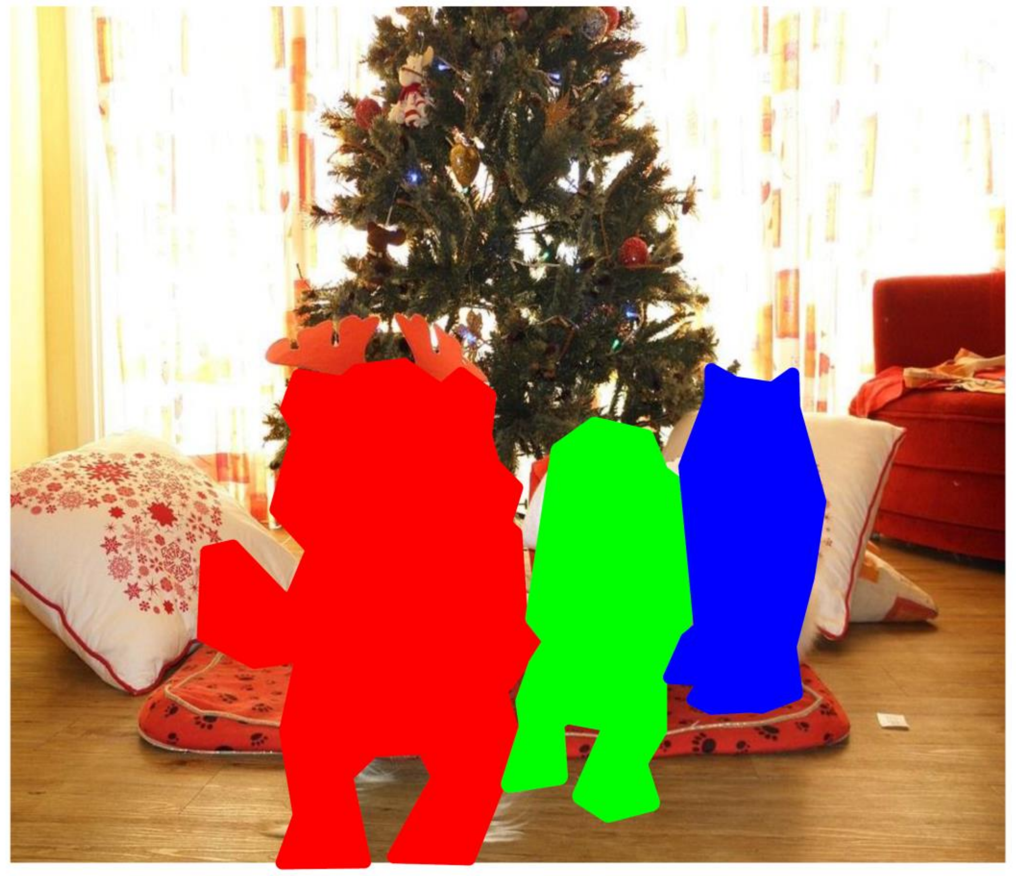
No objects, just pixels

Object Detection



DOG, DOG, CAT

Instance Segmentation



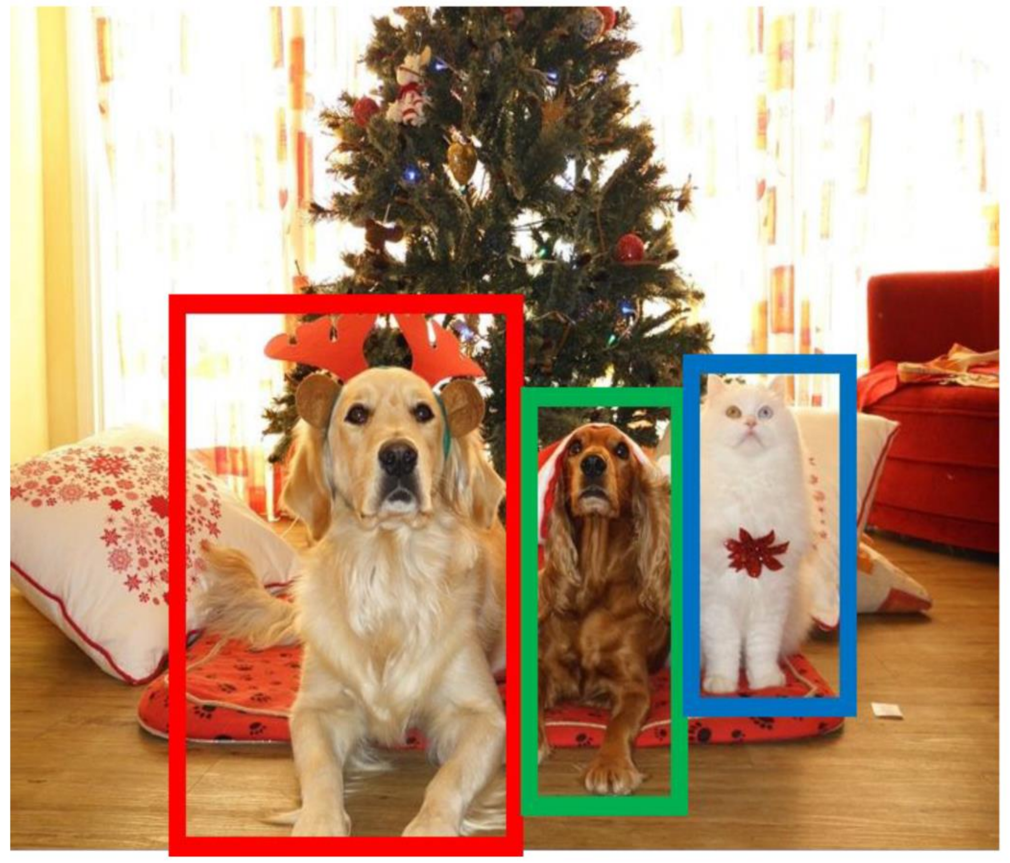
DOG, DOG, CAT

Multiple Object

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Object Detection: *Faster R-CNN*

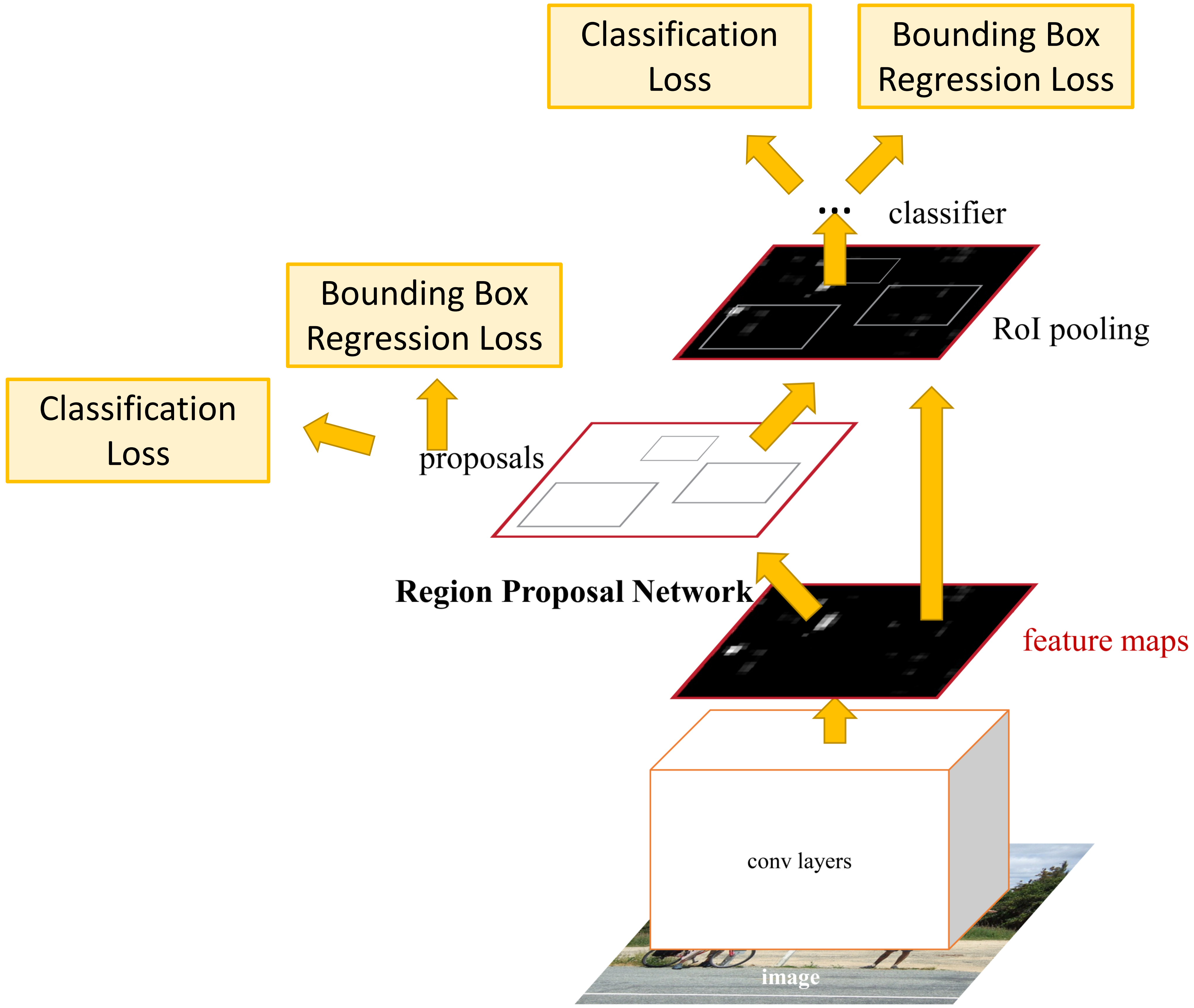
Object Detection



DOG, DOG, CAT

Instance Segmentation

DOG, DOG, CAT

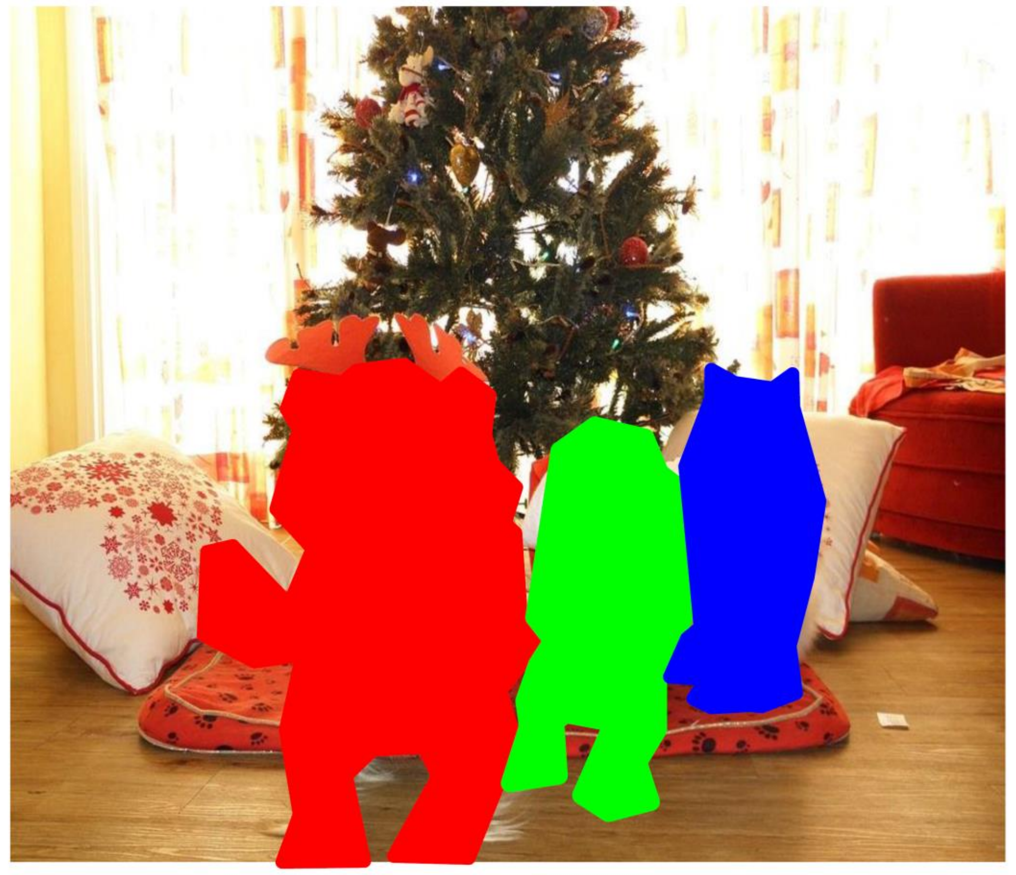


Instance Segmentation: *Mask R-CNN*

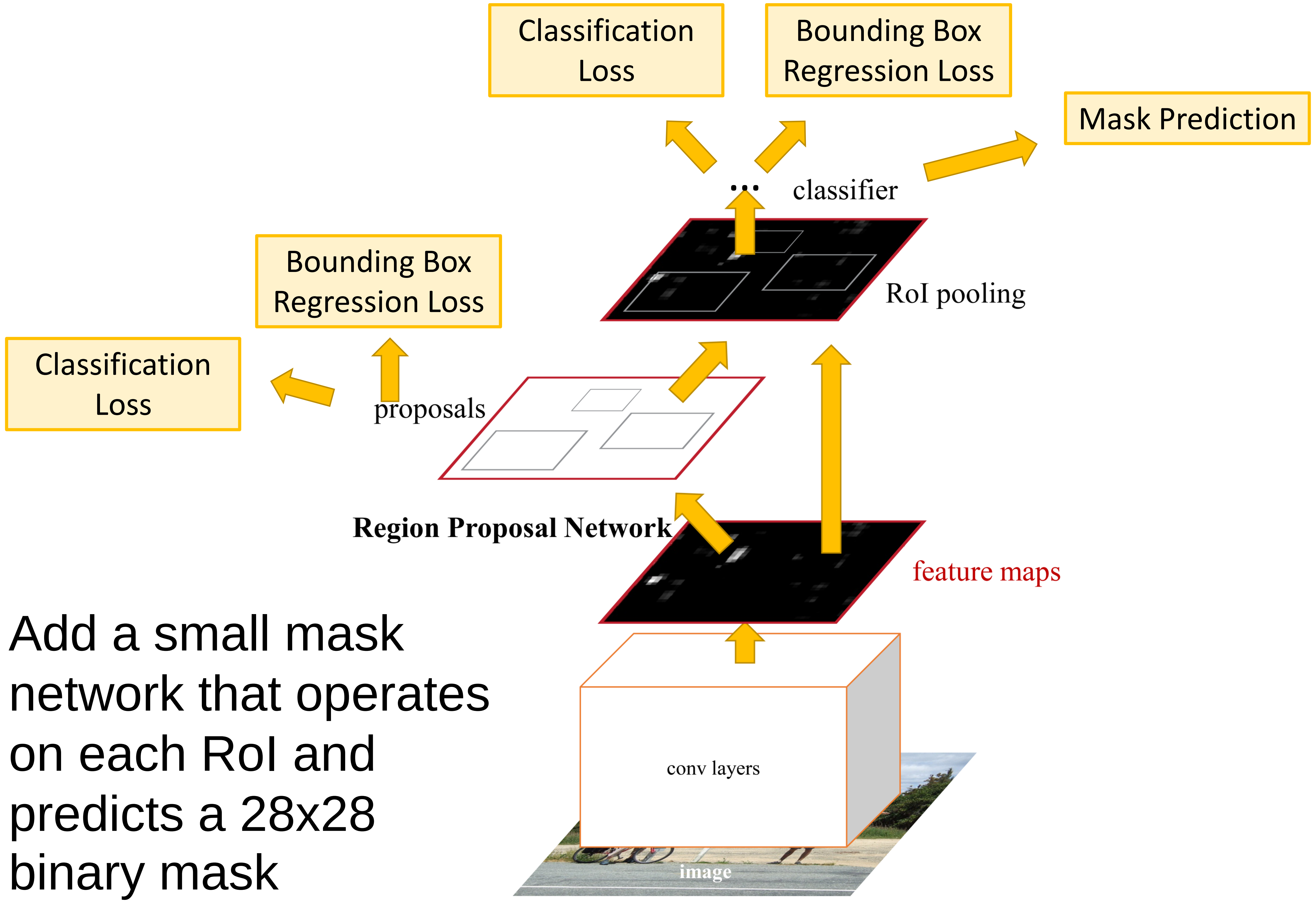
Object Detection

DOG, DOG, CAT

Instance Segmentation



DOG, DOG, CAT



Add a small mask network that operates on each RoI and predicts a 28x28 binary mask

He et al, “Mask R-CNN”, arXiv 2017

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Instance Segmentation: *Mask R-CNN*

Mask R-CNN is a popular object detection and segmentation algorithm that builds upon the earlier Region-based Convolutional Neural Network (R-CNN) and Faster R-CNN object detection algorithms, but with the added capability of instance segmentation.

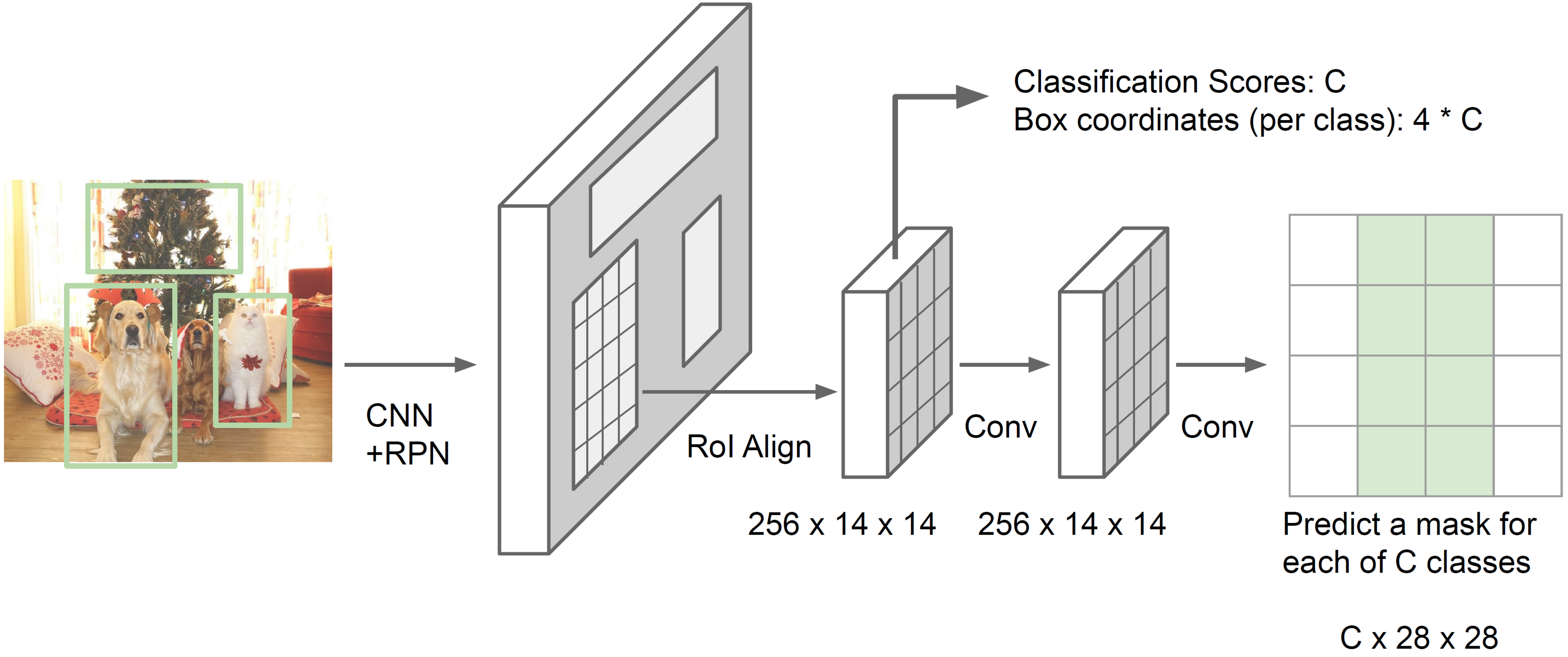
Like its predecessors, Mask R-CNN uses a region proposal network (RPN) to generate candidate object bounding boxes, and a classifier to predict the class and refine the bounding boxes. However, Mask R-CNN also adds a mask branch to the network that predicts binary masks for each object instance, in addition to the bounding box and class labels.

The mask branch takes a cropped feature map of the proposed region and applies a small fully convolutional network to generate a binary mask for the object instance. This allows Mask R-CNN to perform both object detection and instance segmentation in a single pass of the neural network.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

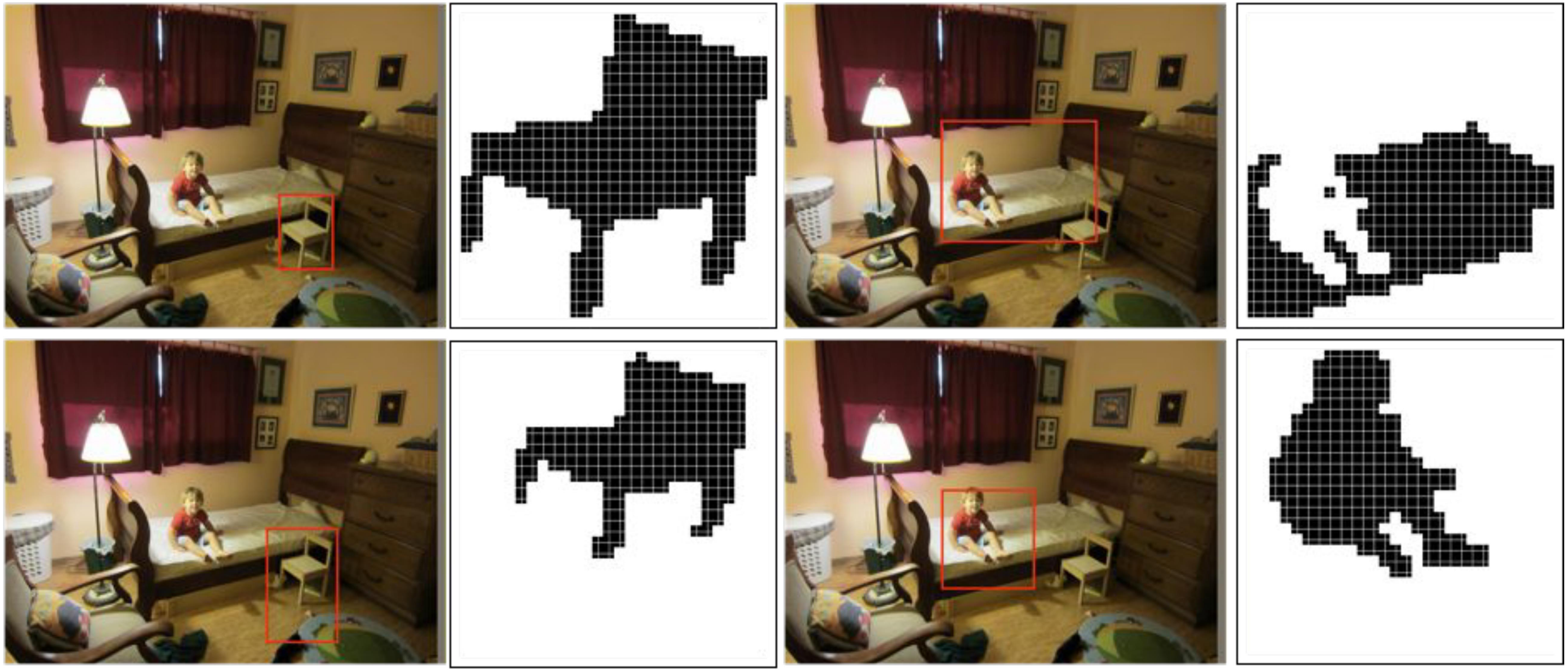
Instance Segmentation: *Mask R-CNN*



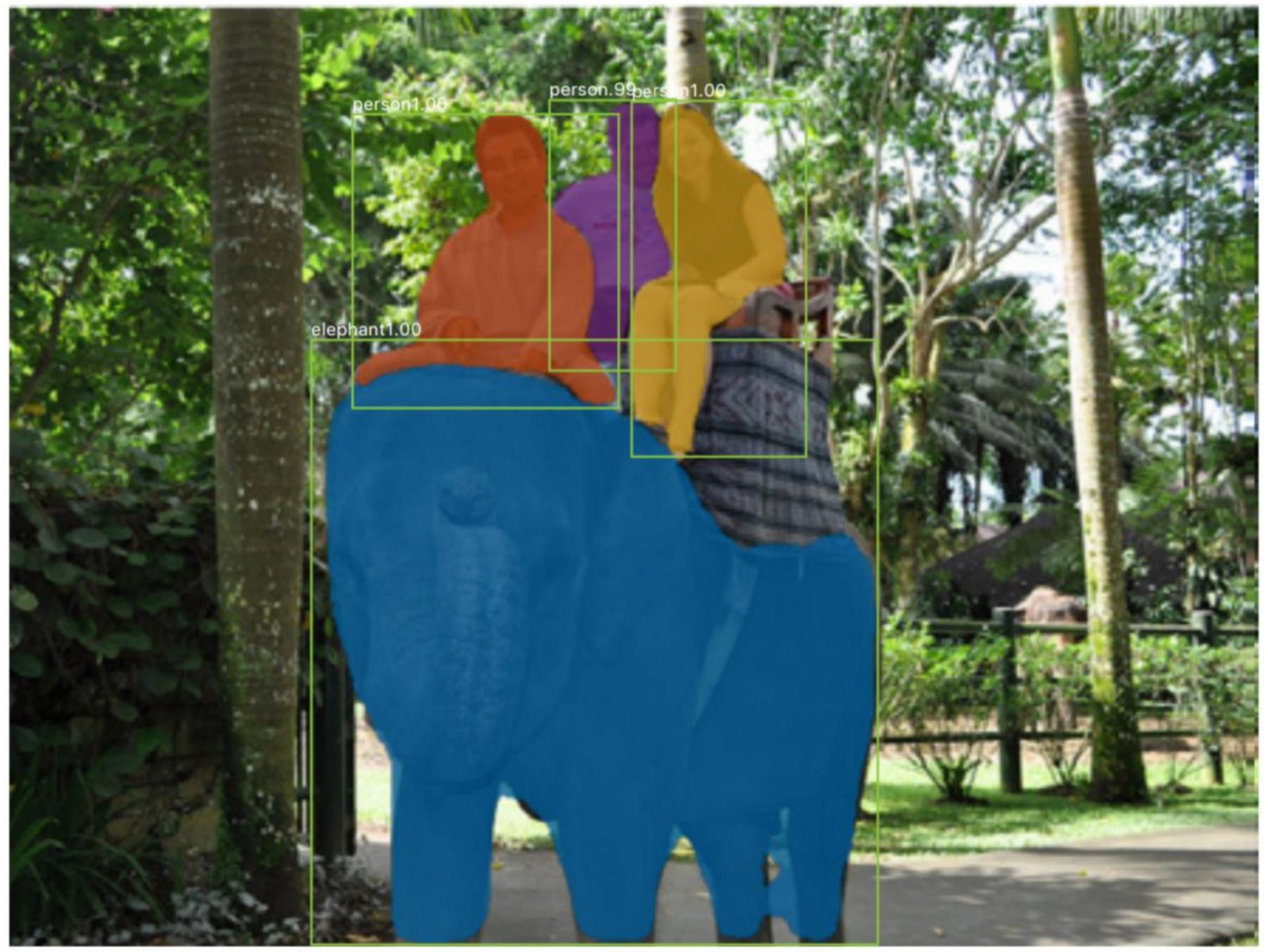
Co-financed by the European Union
Connecting Europe Facility

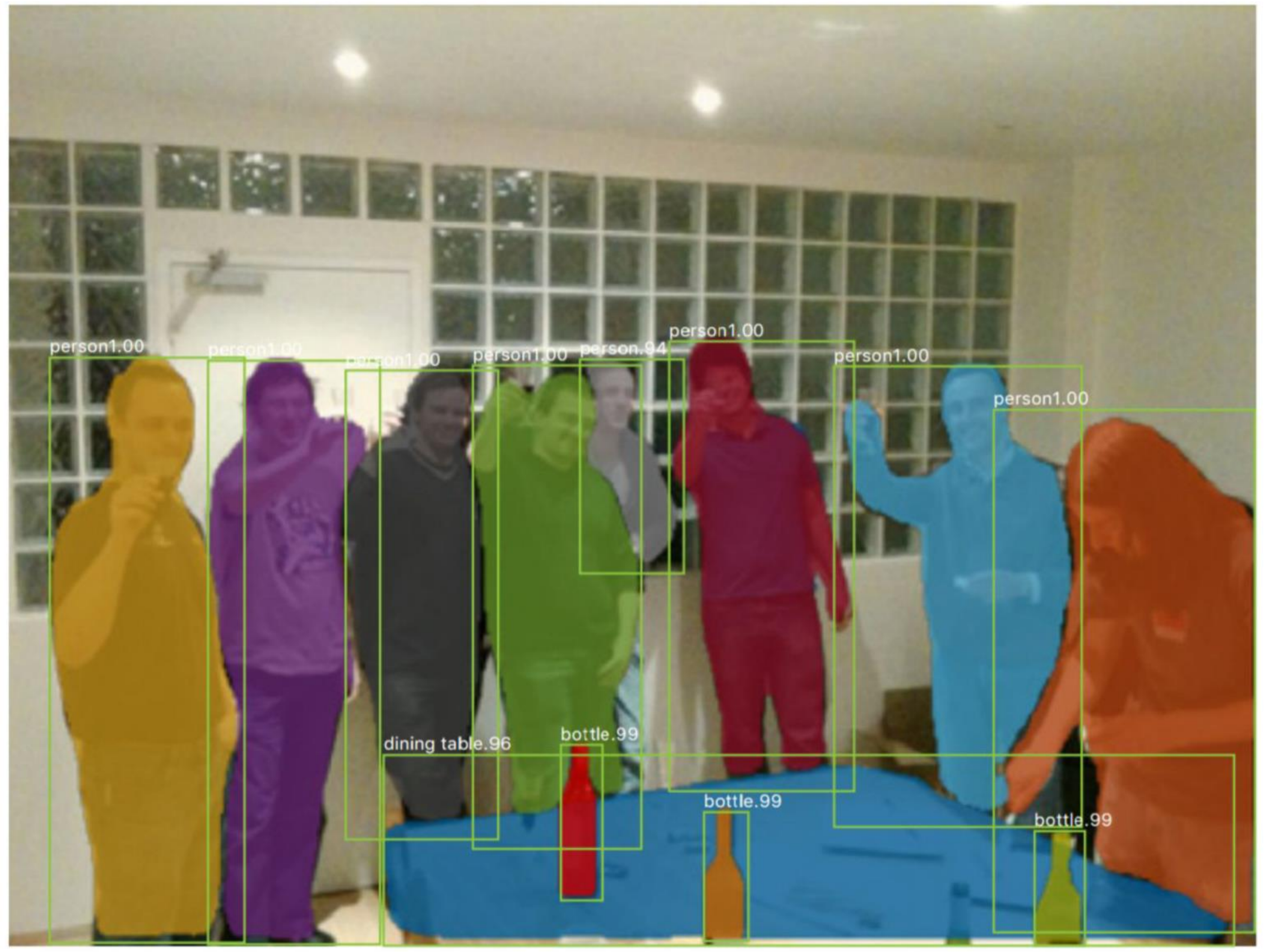
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Instance Segmentation: *Mask R-CNN*



Instance Segmentation: *Mask R-CNN - Very Good Results!*





Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Instance Segmentation: *Mask R-CNN – Also does pose!*

Open Source Frameworks

Lots of good implementations on GitHub!

TensorFlow Detection API:
https://github.com/tensorflow/models/tree/master/research/object_detection
Faster RCNN, SSD, RFCN, Mask R-CNN, ...

Detectron2 (PyTorch)
https://github.com/facebookresearch/detectron2
Mask R-CNN, RetinaNet, Faster R-CNN, RPN, Fast R-CNN, R-FCN, ...

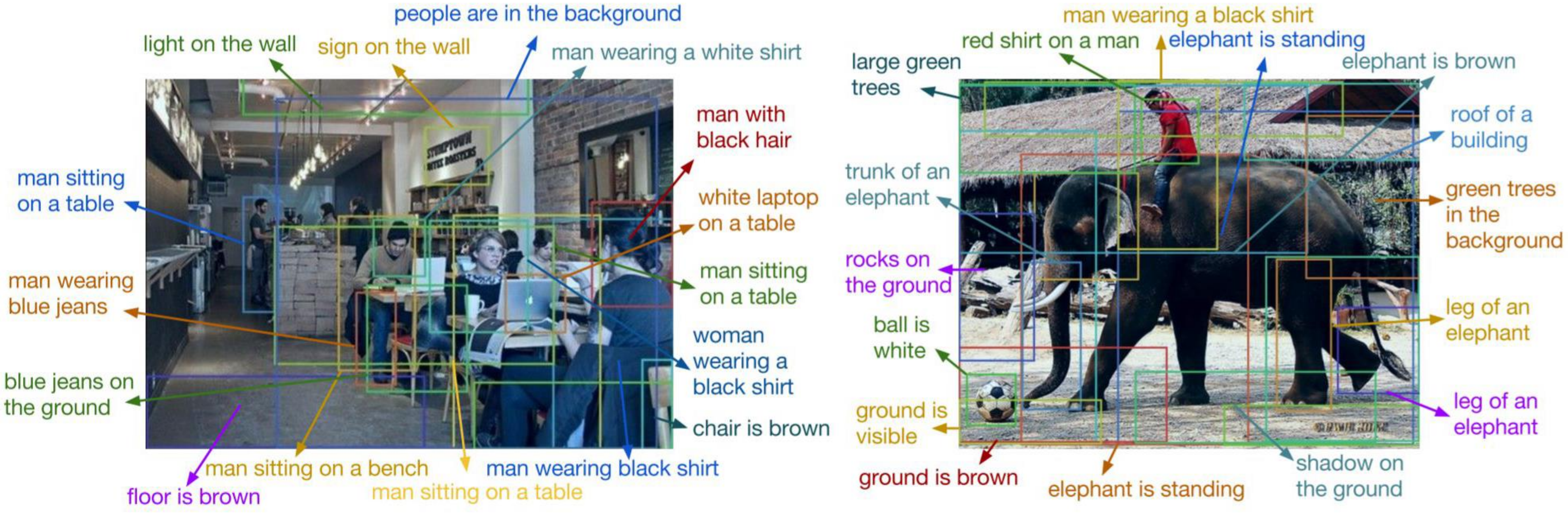
Finetune on your own dataset with pre-trained models

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Beyond 2D Object Detection...

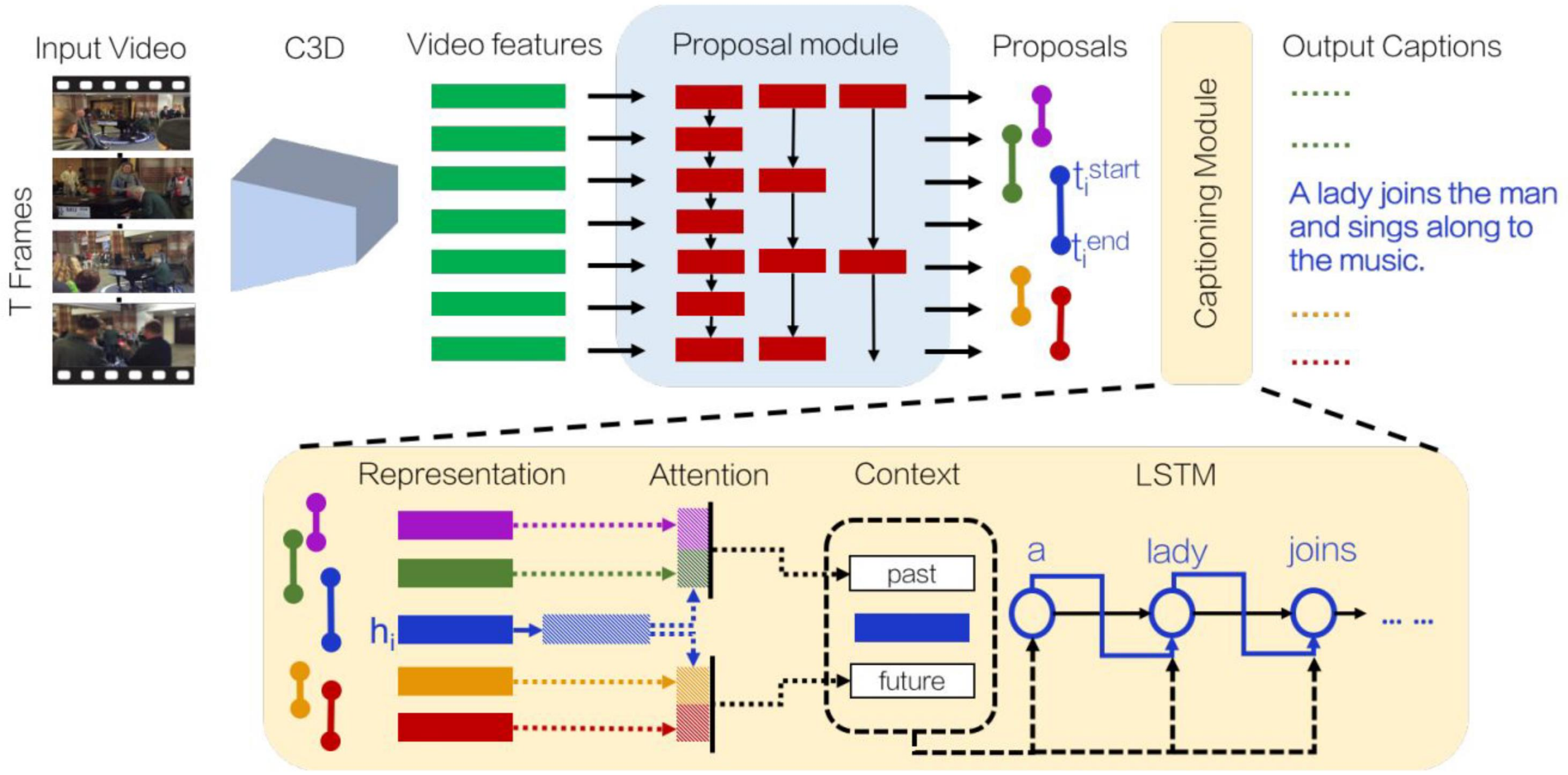
Object Detection + Captioning = Dense Captioning

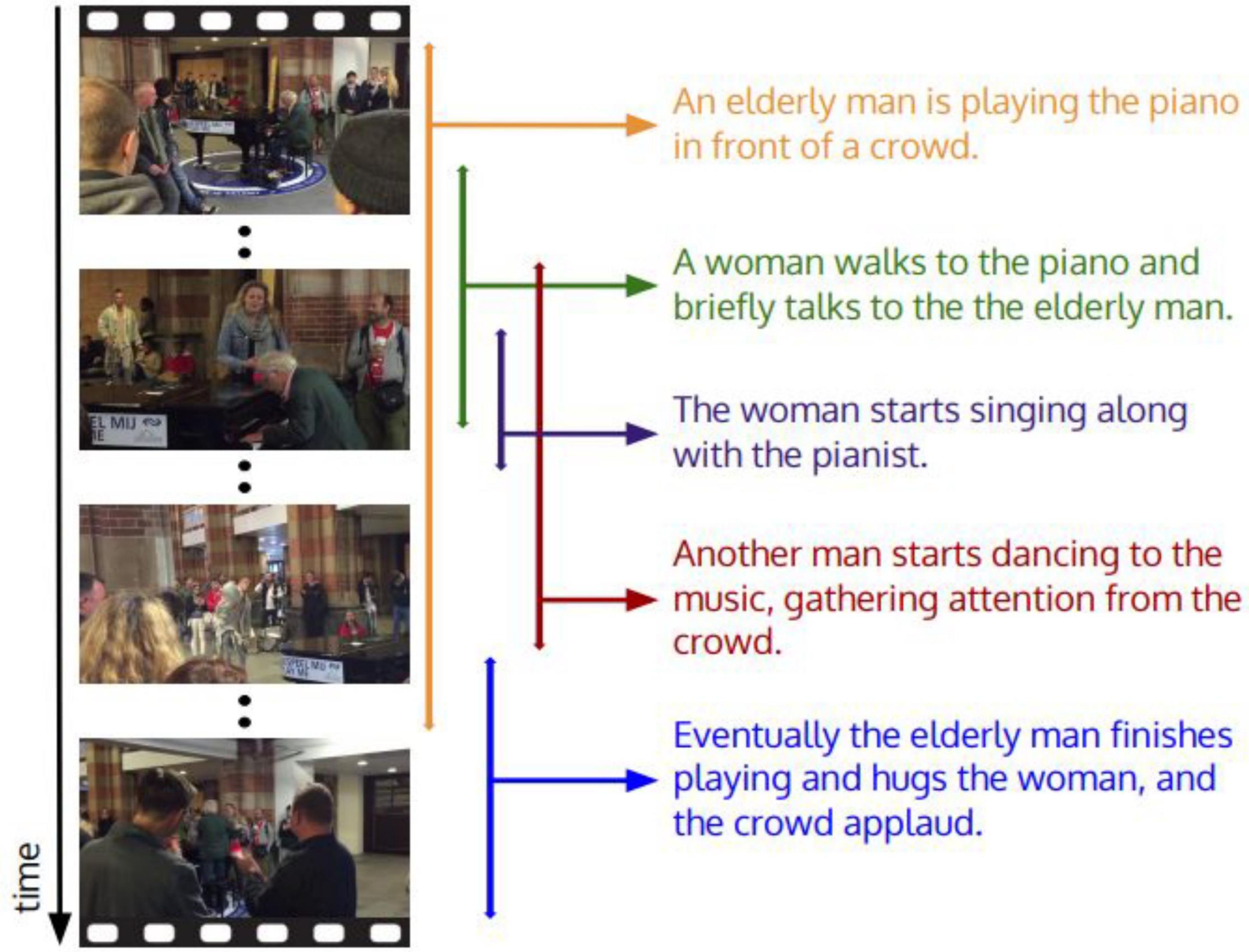


Johnson, Karpathy, and Fei-Fei, “DenseCap: Fully Convolutional Localization Networks for Dense Captioning”, CVPR 2016

Beyond 2D Object Detection...

Dense Video Captioning

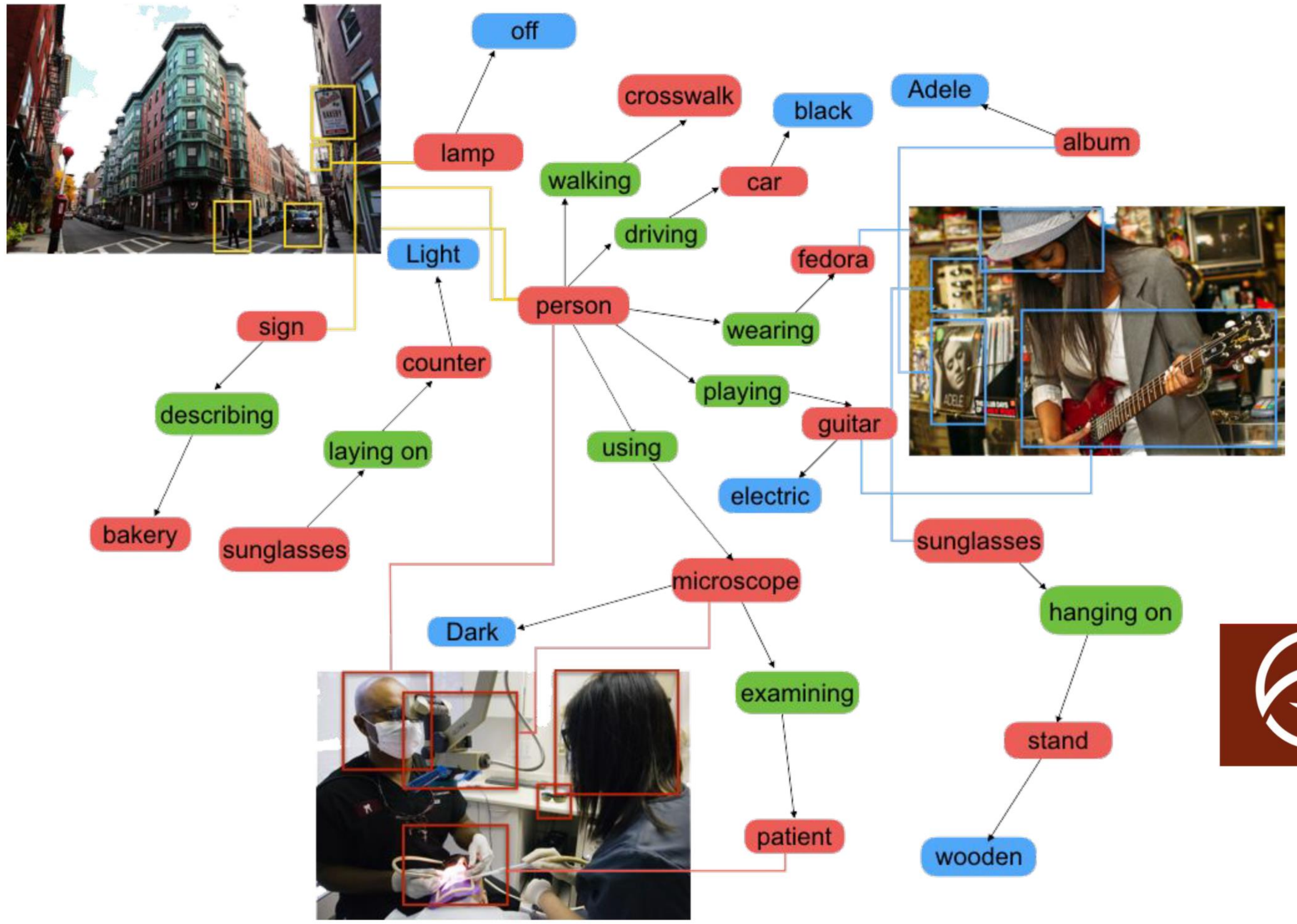




Ranjay Krishna et al., “Dense-Captioning Events in Videos”, ICCV 2017

Beyond 2D Object Detection...

Objects + Relationships = Scene Graphs

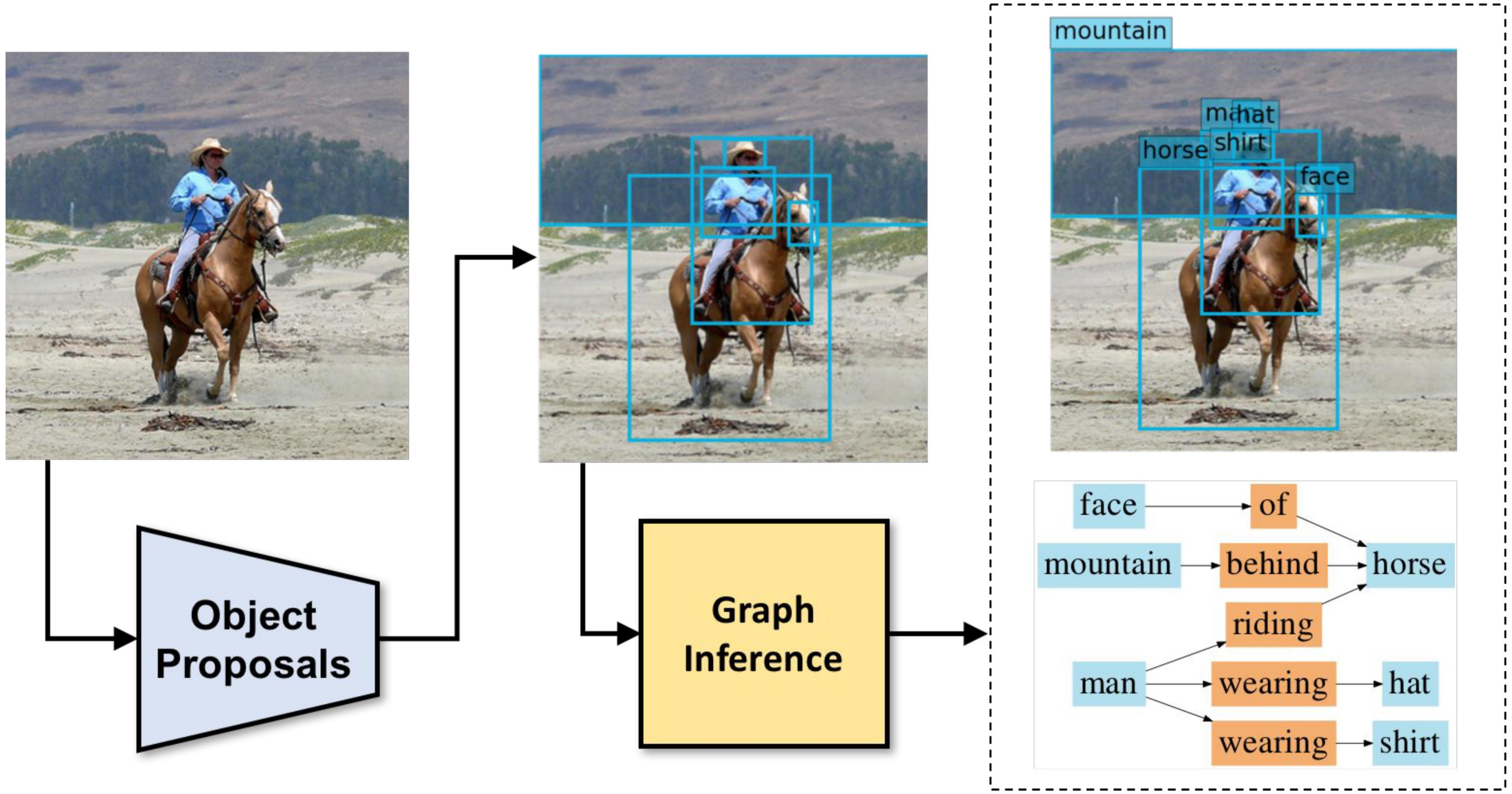


VISUALGENOME

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen et al. "Visual genome: Connecting language and vision using crowdsourced dense image annotations." International Journal of Computer Vision 123, no. 1 (2017): 32-73.

Beyond 2D Object Detection...

Scene Graph Prediction

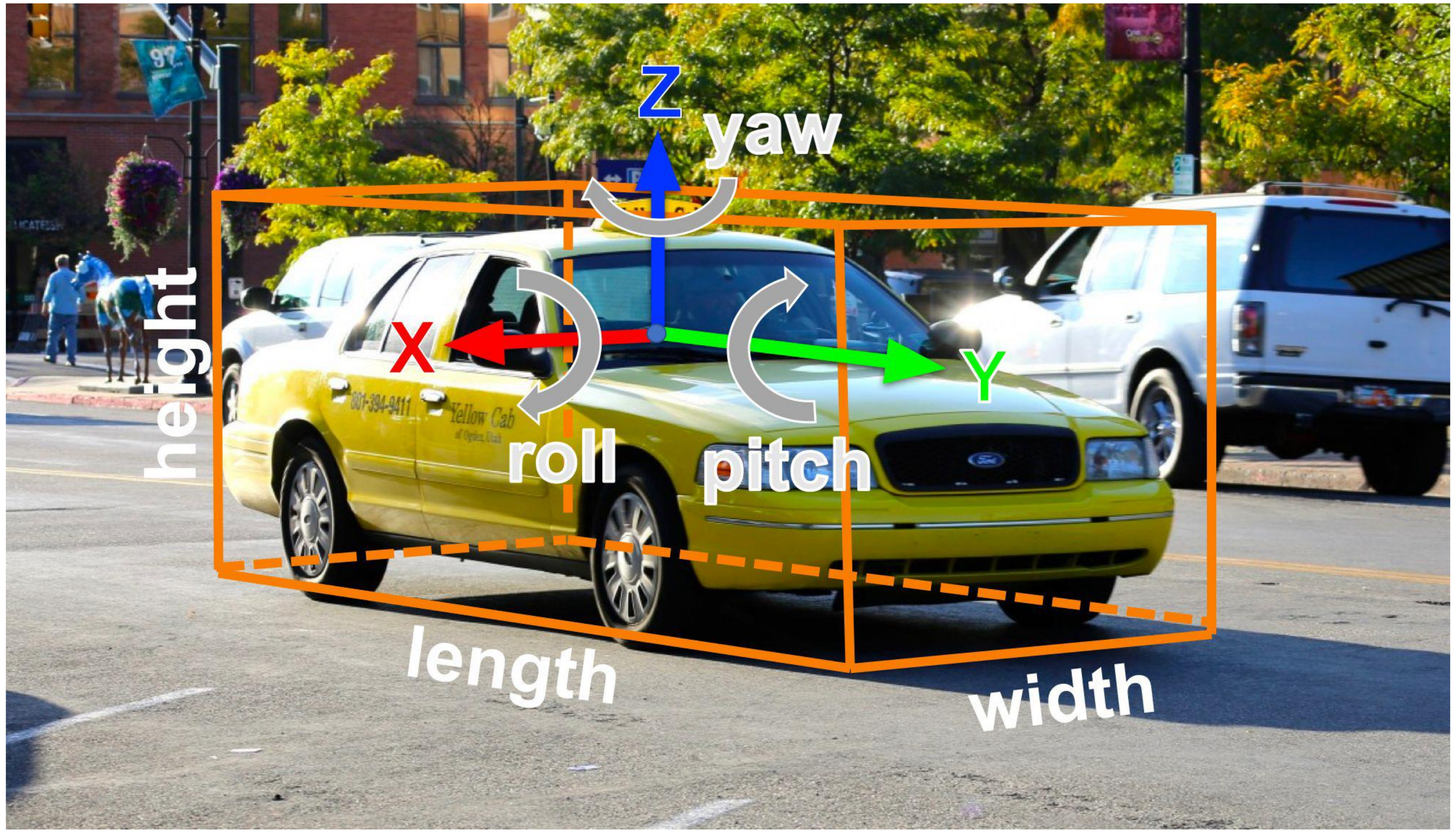


Xu, Zhu, Choy, and Fei-Fei, “Scene Graph Generation by Iterative Message Passing”, CVPR 2017

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Beyond 2D Object Detection...

3D Object Detection



2D Object Detection:
2D bounding box
(x, y, w, h)

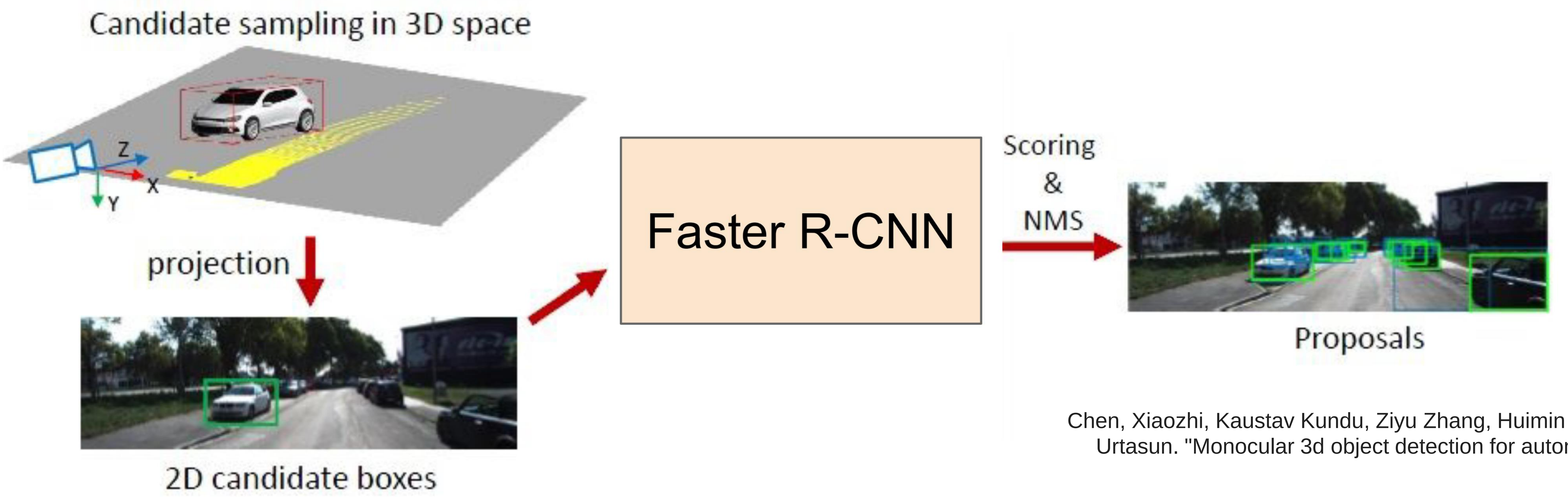
3D Object Detection:
3D oriented bounding box
(x, y, z, w, h, l, r, p, y)

Simplified bbox: no roll & pitch

Much harder problem than 2D object detection!

Beyond 2D Object Detection...

3D Object Detection: Monocular Camera



Chen, Xiaozhi, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. "Monocular 3d object detection for autonomous driving." CVPR 2016.

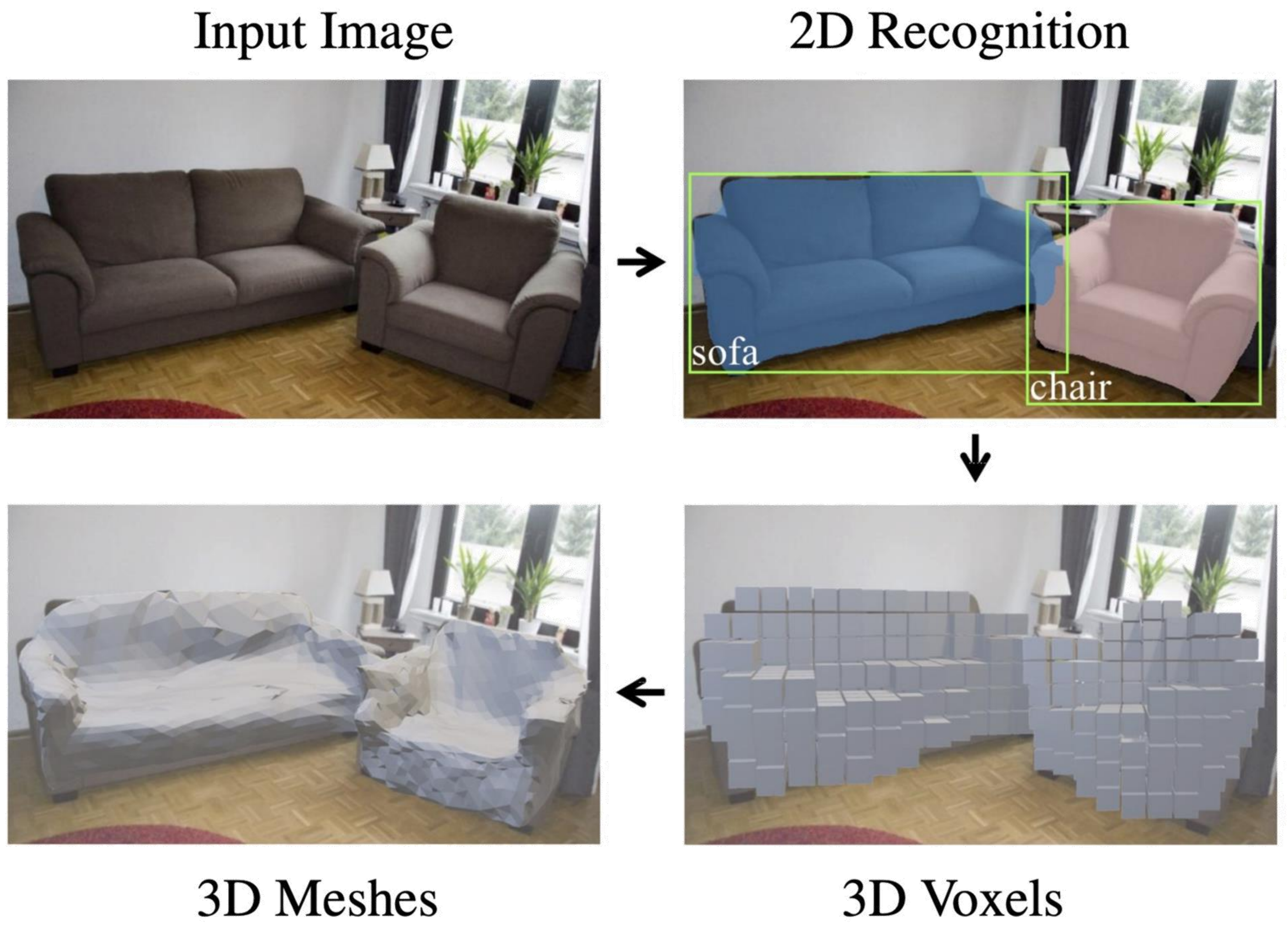
- Same idea as Faster RCNN, but proposals are in 3D
- 3D bounding box proposal, regress 3D box parameters + class score

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Beyond 2D Object Detection...

3D Shape Prediction: Mesh R-CNN



Gkioxari et al., Mesh RCNN, ICCV 2019

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Research Example from DeepCamera, CYENS Centre of Excellence

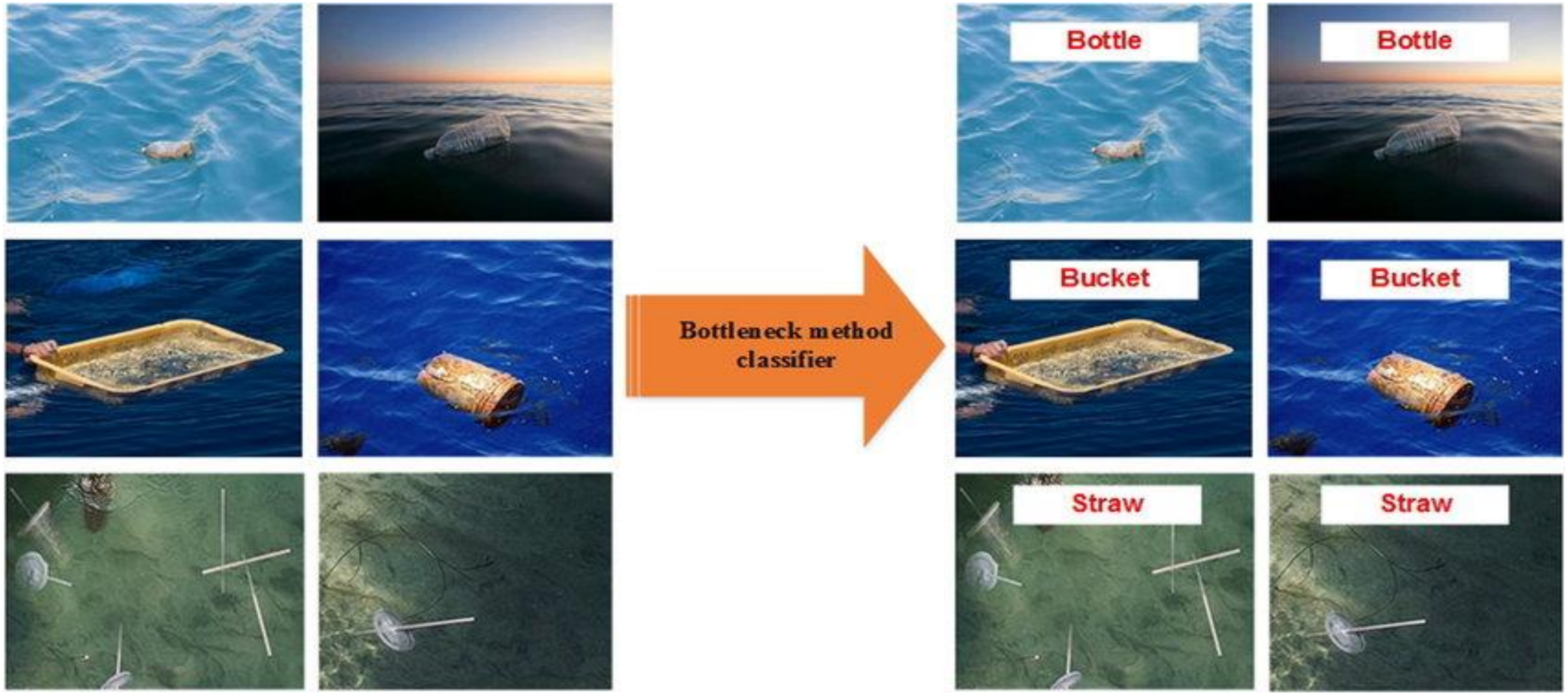
CYENS
CENTRE OF EXCELLENCE

Deep
Camera

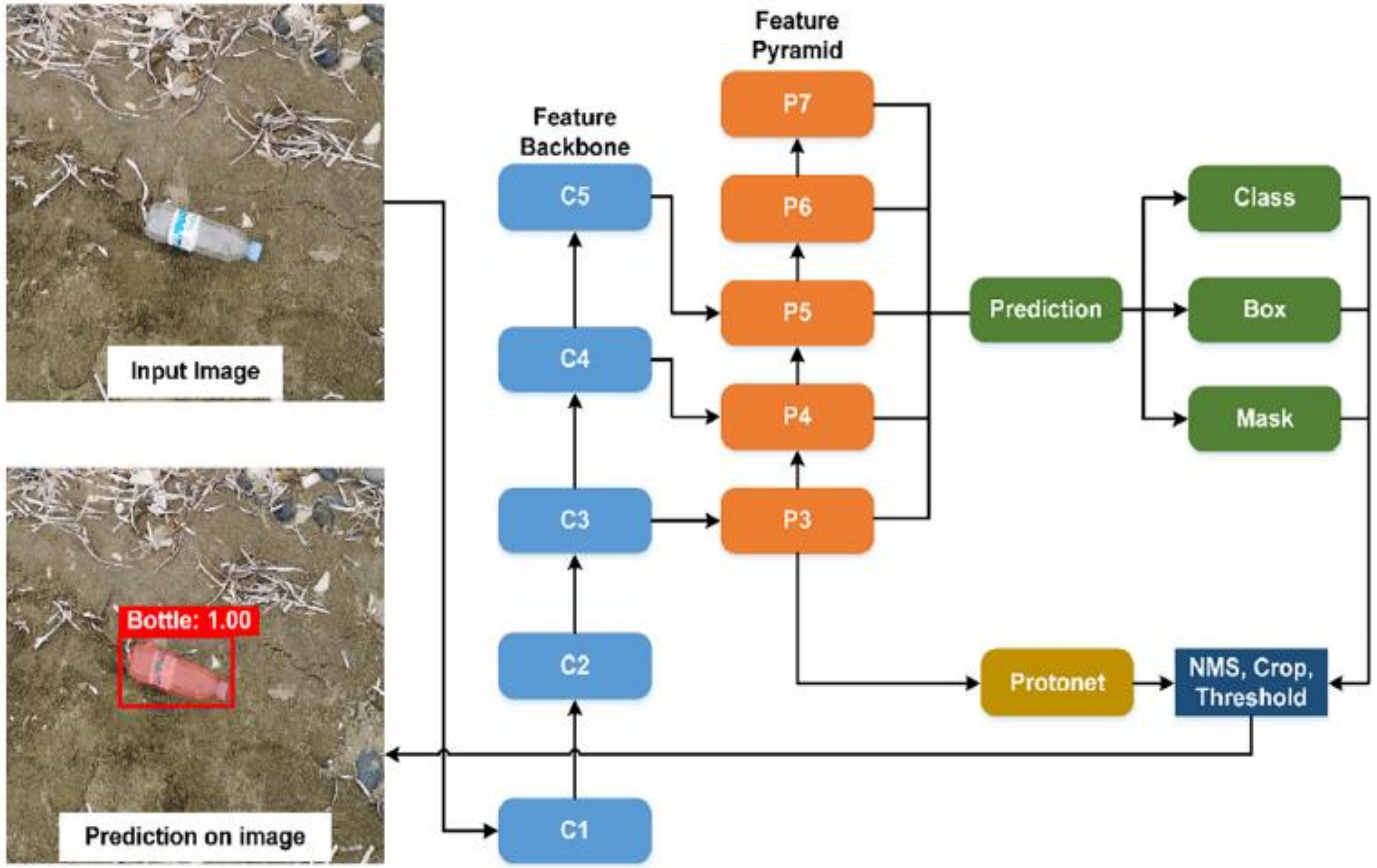
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Identifying floating plastic marine debris using a deep learning approach

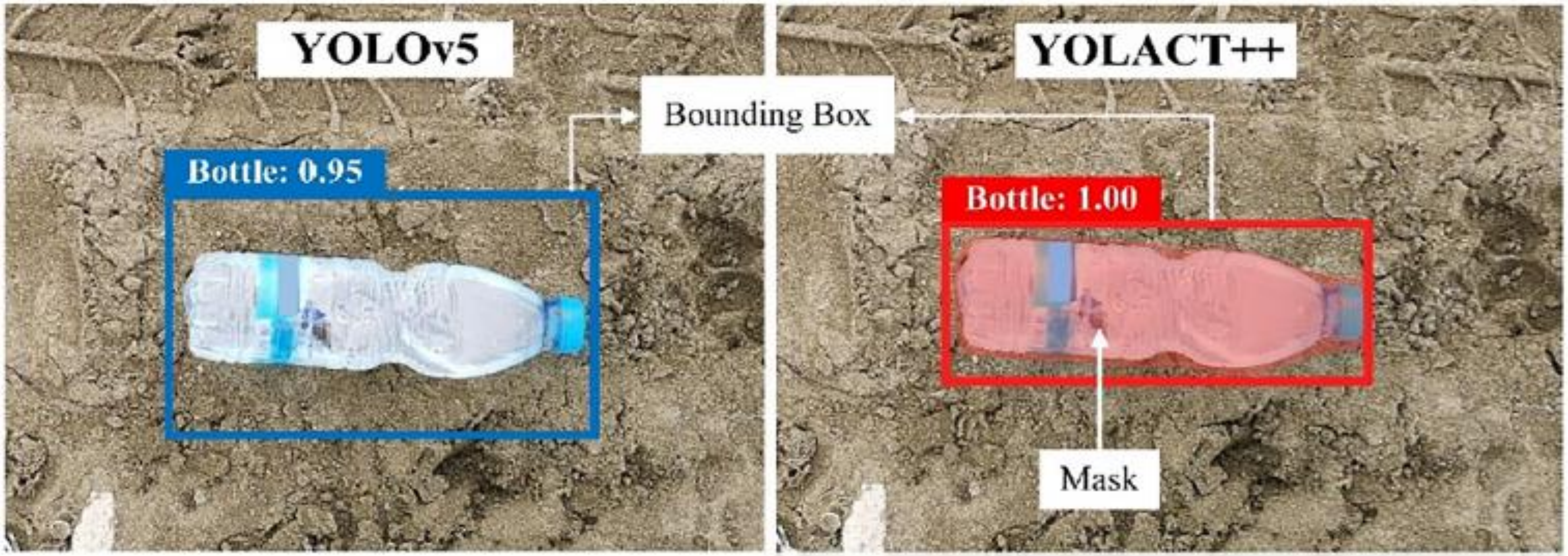


A new paradigm for estimating the prevalence of plastic litter in the marine environment

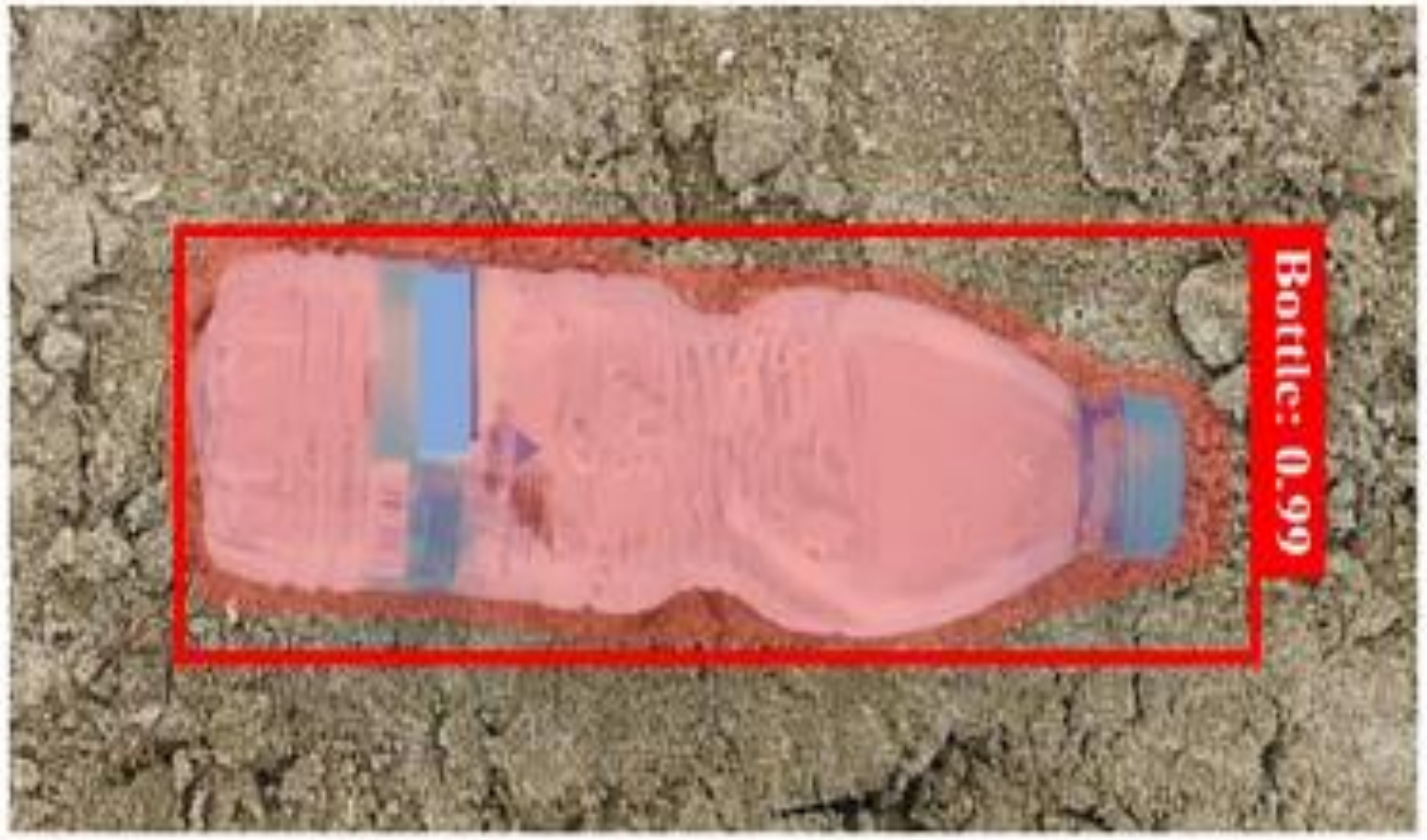


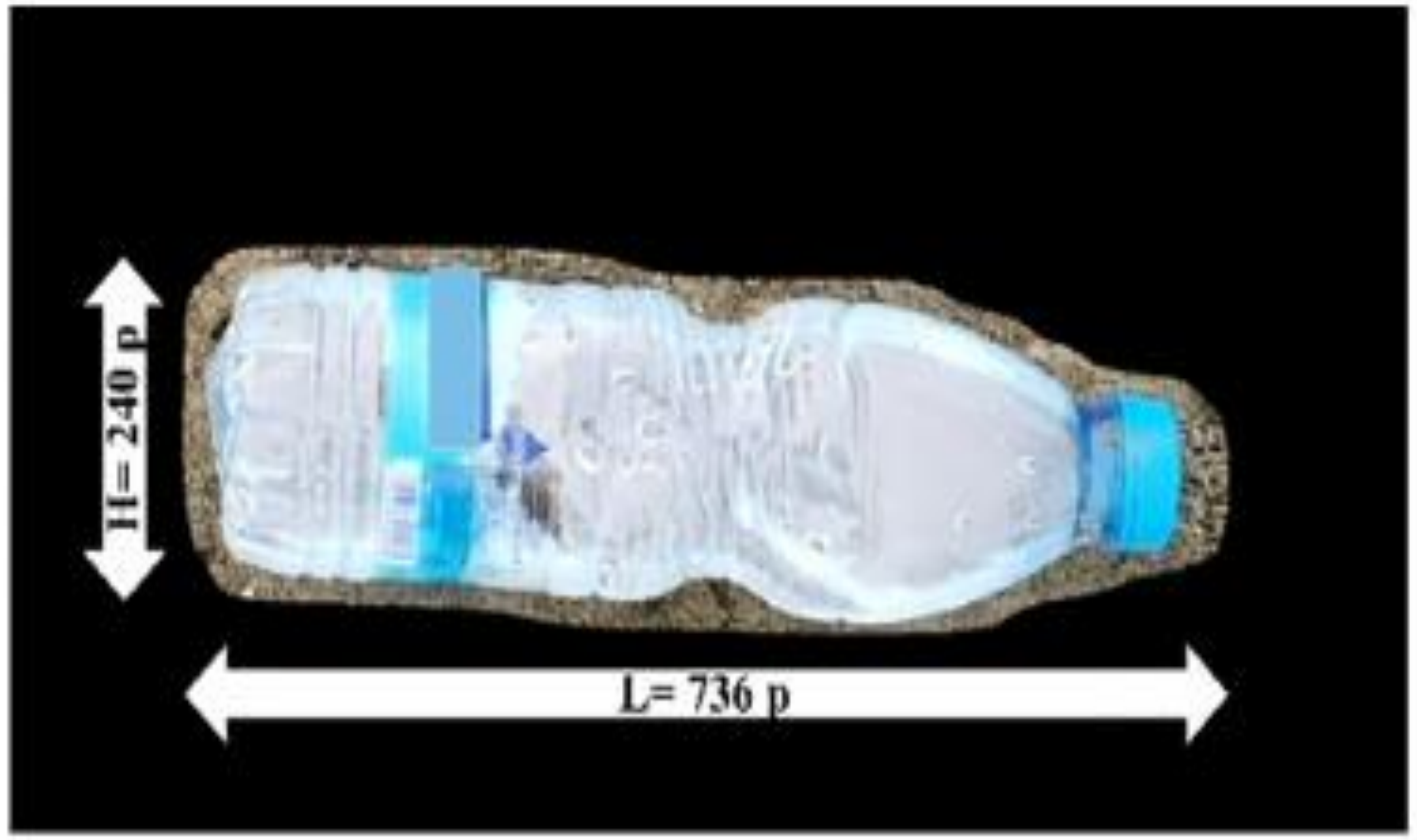
The YOLACT++ model architecture

Comparison of the performance of the YOLOv5 and the YOLACT++ tools



Determining the dimensions of plastic litter





Research in Deep Camera

Alessandro Artusi
Team Leader
DeepCamera Group

email: a.artusi@cyens.org.cy

Research Interests:

Machine Learning, Deep Learning and its applications in Computer Vision, High Dynamic Range Imaging, Image Processing applied on Computer Graphics and Color Science

https://www.cyens.org.cy/en-gb/research/pillars-groups/visual-sciences/deep-camera/people/alessandro-artusi/

Deep Camera

CYENS
CENTRE OF EXCELLENCE

https://deepcamera.cyens.org.cy/about-us/

deepcamera.ai@gmail.com

+357 227 475 81

Dimarchias Square 23 STOA, Nicosia
Nicosia, Nicosia 1016, Cyprus

Thank you!

See you next week

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423