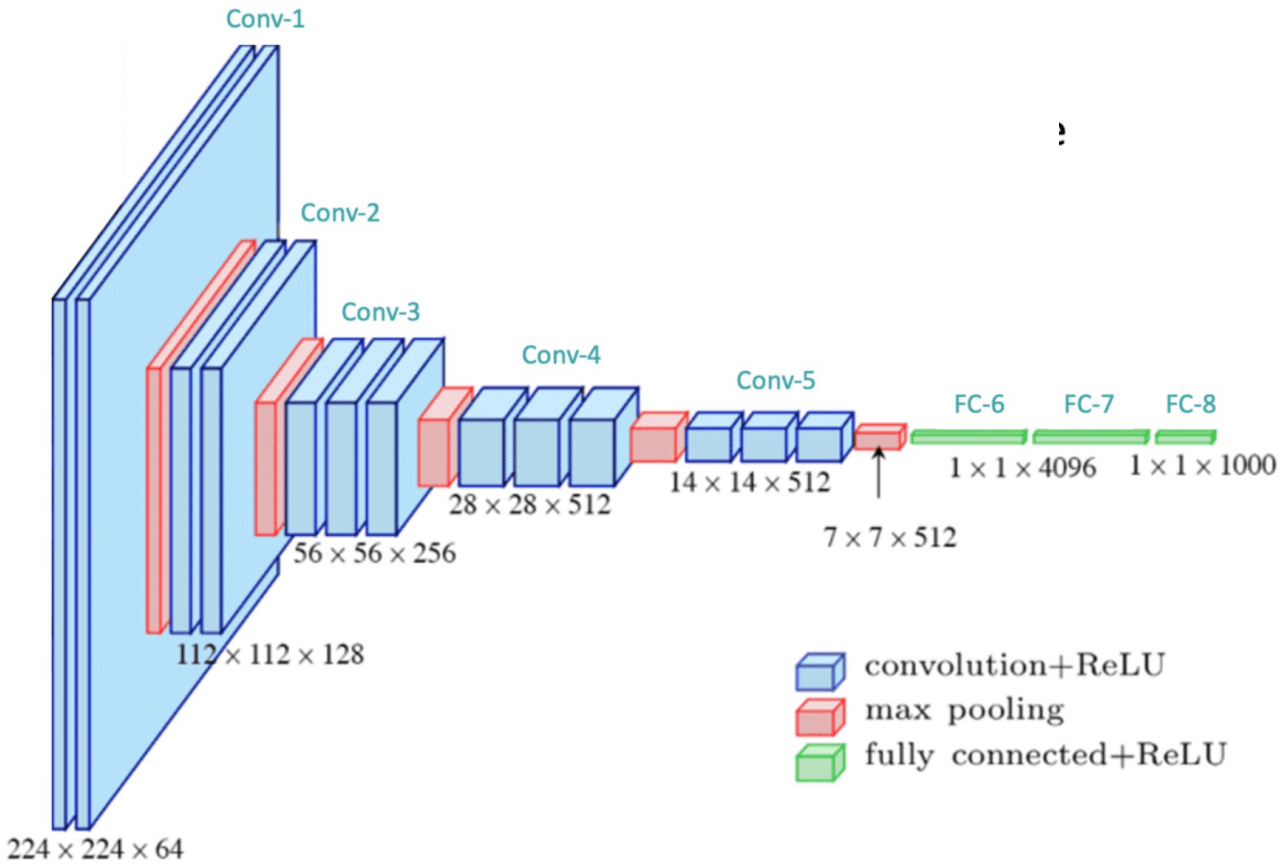
Lecture 3.1

Object Detection

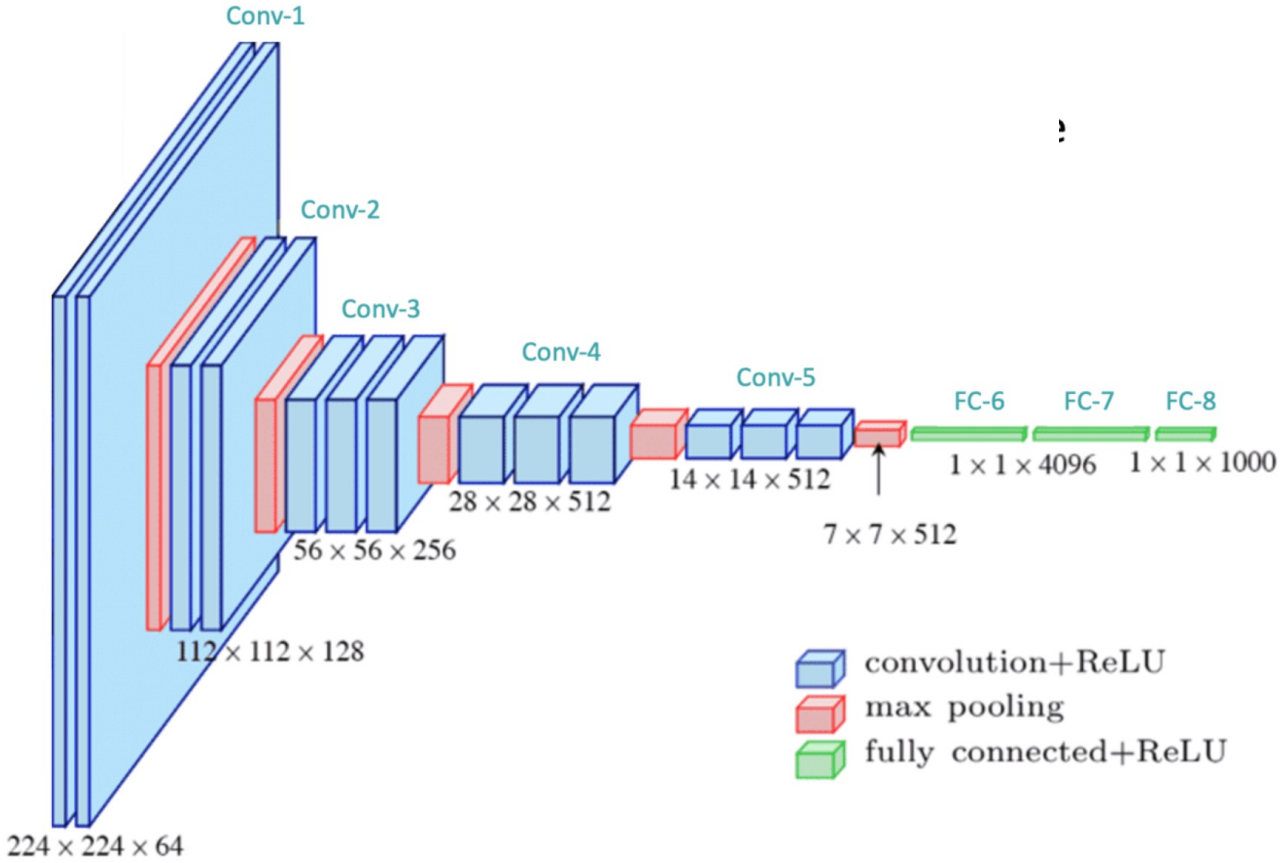
Dimitrios Papadopoulos
Associate Professor, DTU Compute

So far, you have learned…

So far, you have learned…

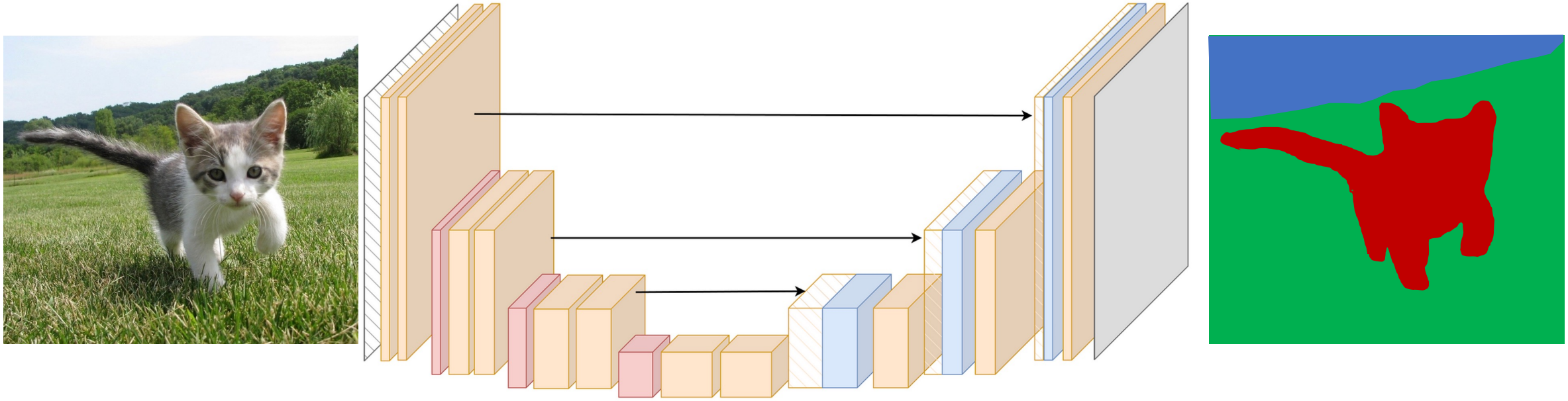


So far, you have learned…



cat

So far, you have learned…



Object Detection

Part 3: Object Detection

Wednesday 30.10	13:00-15:00 Lecture - Object Detection 15:00-17:00 Exercises	Dimitrios TAs Marco and Paraskevas
Wednesday 6.11	13:00-14:30 Lecture - Object Detection 14:30 - 16:00 - **Poster session** 16:00-17:00 Exercises	Dimitrios Dimitrios, Aasa, TAs Marco, Changlu, Paraskevas TAs Changlu and Marco
Wednesday 13.11	13:00-15:00 Lecture - Object Detection 15:00-17:00 Exercises	Dimitrios TAs Changlu and Marco
Tuesday 19.11	**Poster submission at 22:00**	

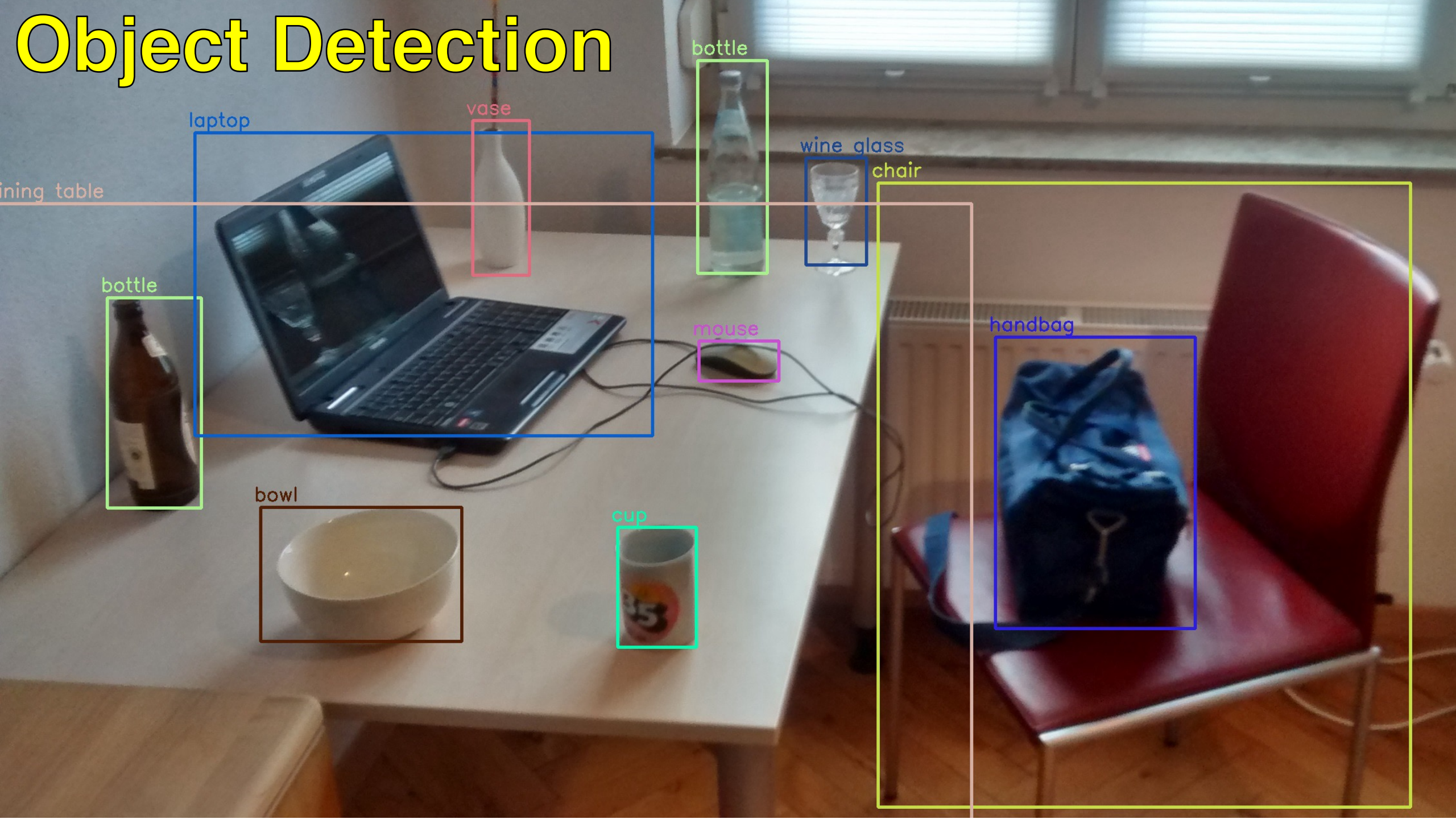
Object Detection

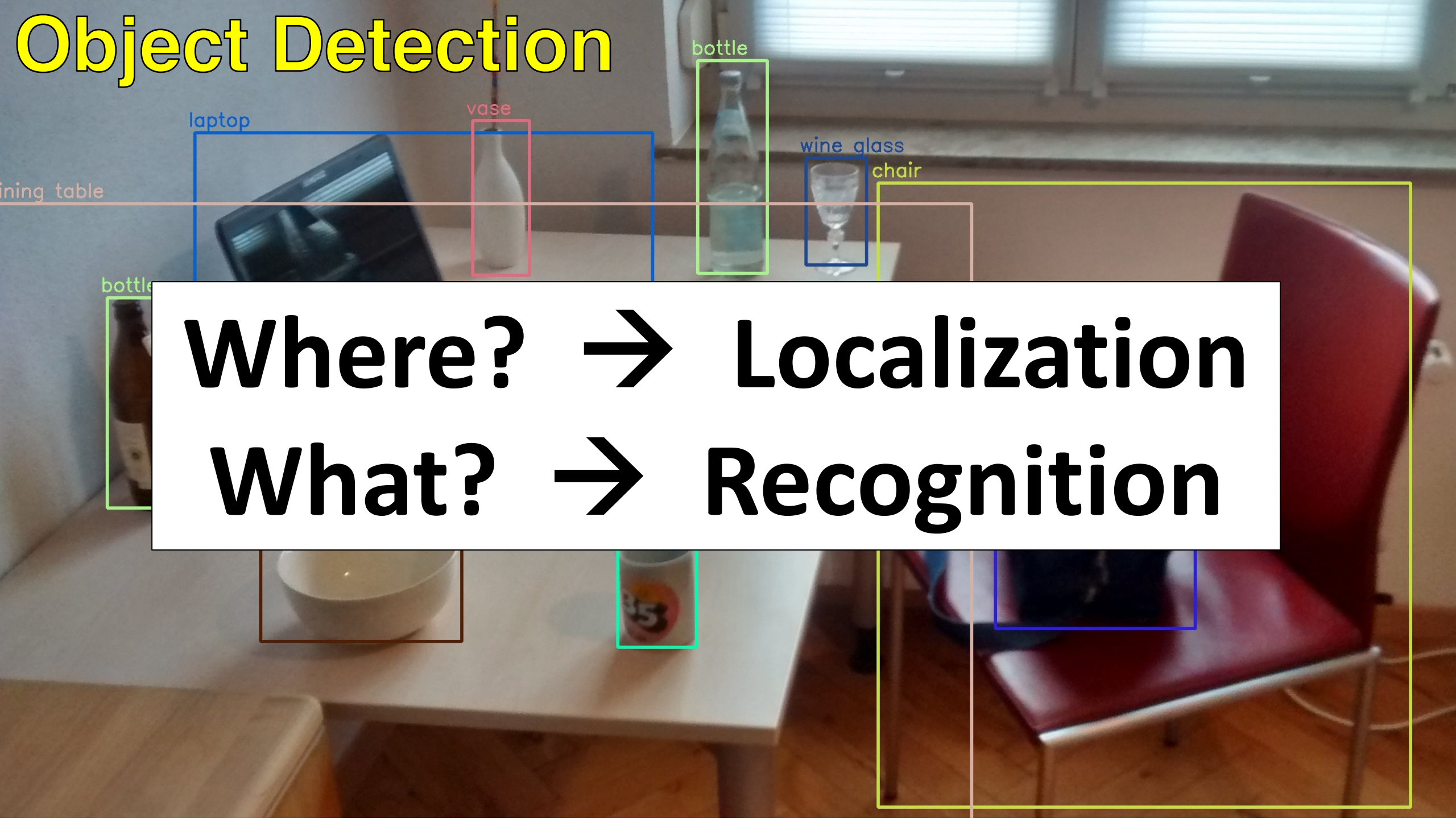
Part 3: Object Detection

Friday 12.1	09:00-11:00 Lecture - Object Detection 11:00-12:00 Exercises 13:00-17:00 Exercises	Dimitrios Dimitrios TA 14-16
Monday 15.1	09:00-11:00 Lecture - Object Detection 11:00-12:00 Exercises 13:00-14:00 **Poster session** 14:00-17:00 Exercises	Dimitrios Dimitrios TA 14-16
Tuesday 16.1	09:00-11:00 Lecture - Object Detection 11:00-12:00 Exercises 13:00-17:00 Exercises	Dimitrios Dimitrios TA 14-16
Wednesday 17.1	09:00-11:00 Lecture - Object Detection 11:00-12:00 Exercises 13:00-17:00 Exercises	Dimitrios Dimitrios TA 14-16

Today – Object Detection

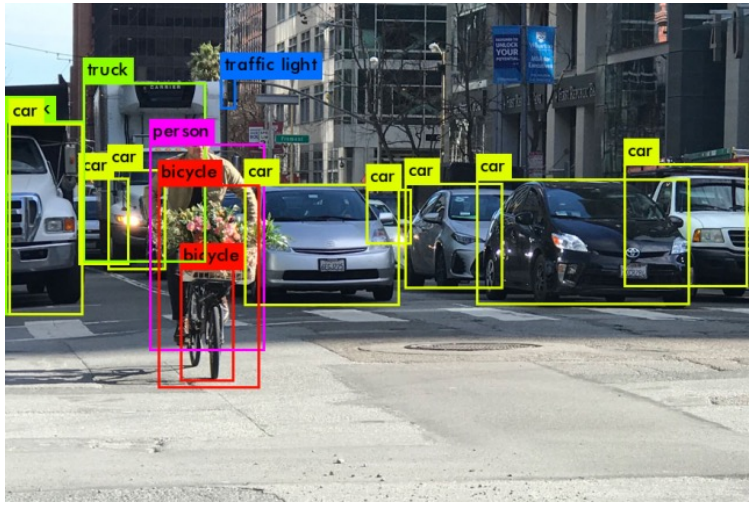
- Why object detection?
- Problem Formulations and General Strategies
- The History of Object Detection (2001 – 2015)
- Object proposals and R-CNN
- **Next weeks:** Fast R-CNN, Faster R-CNN, comparing Boxes and evaluating object detectors, Single-stage detectors (e.g. YOLO), state-of-the-art object detectors


Object Detection
bottle
vase
laptop
wine glass
chair
ining table
bottle
mouse
handbag
bowl
cup

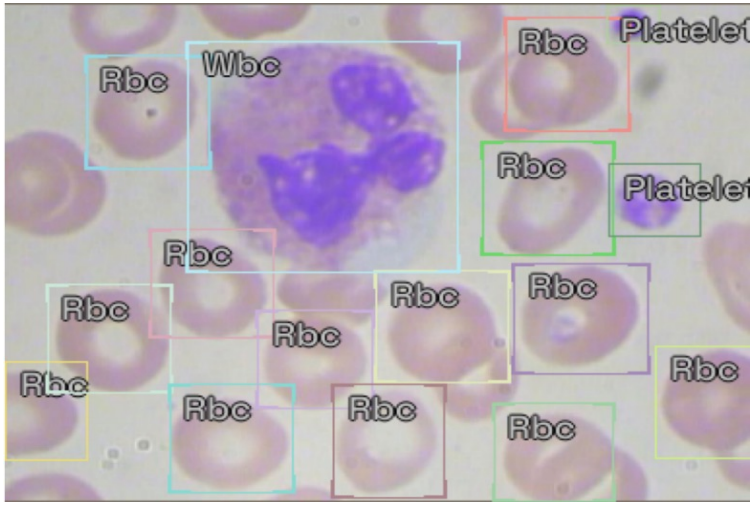

Object Detection
bottle
laptop
vase
wine glass
chair
ining table
bottle
Where? → Localization
What? → Recognition

Object detection applications

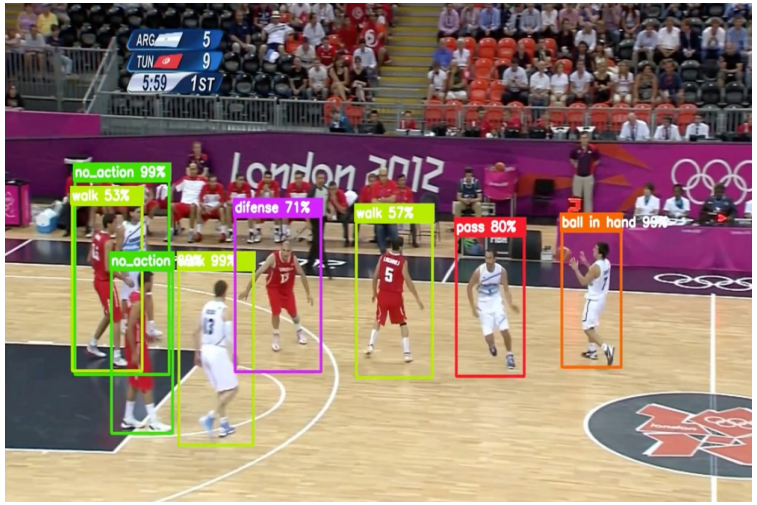
Autonomous driving



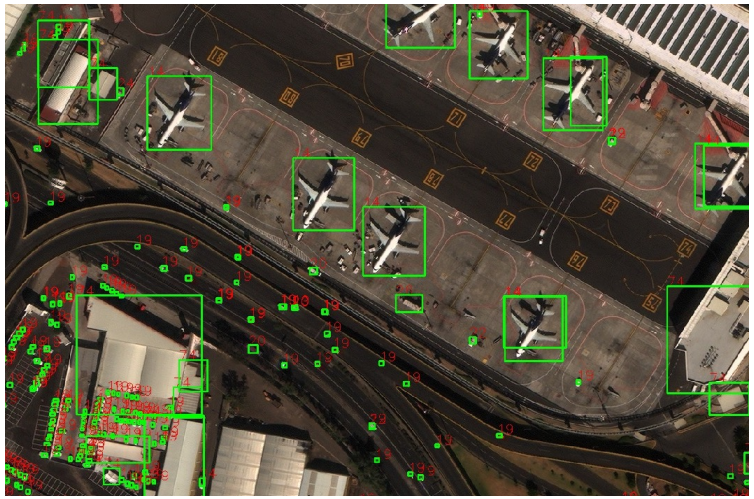
Healthcare



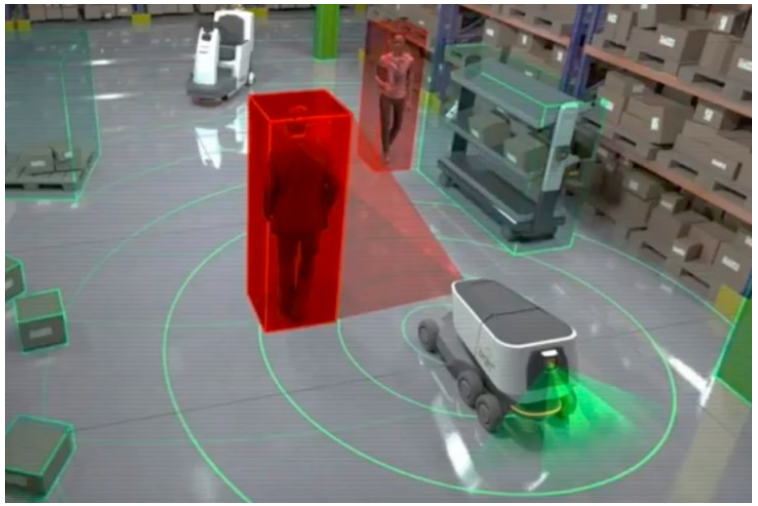
Sport analytics



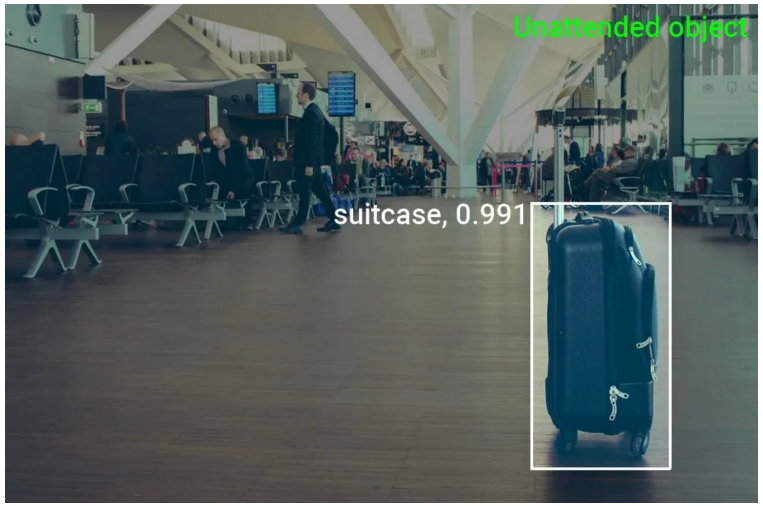
Satellite images



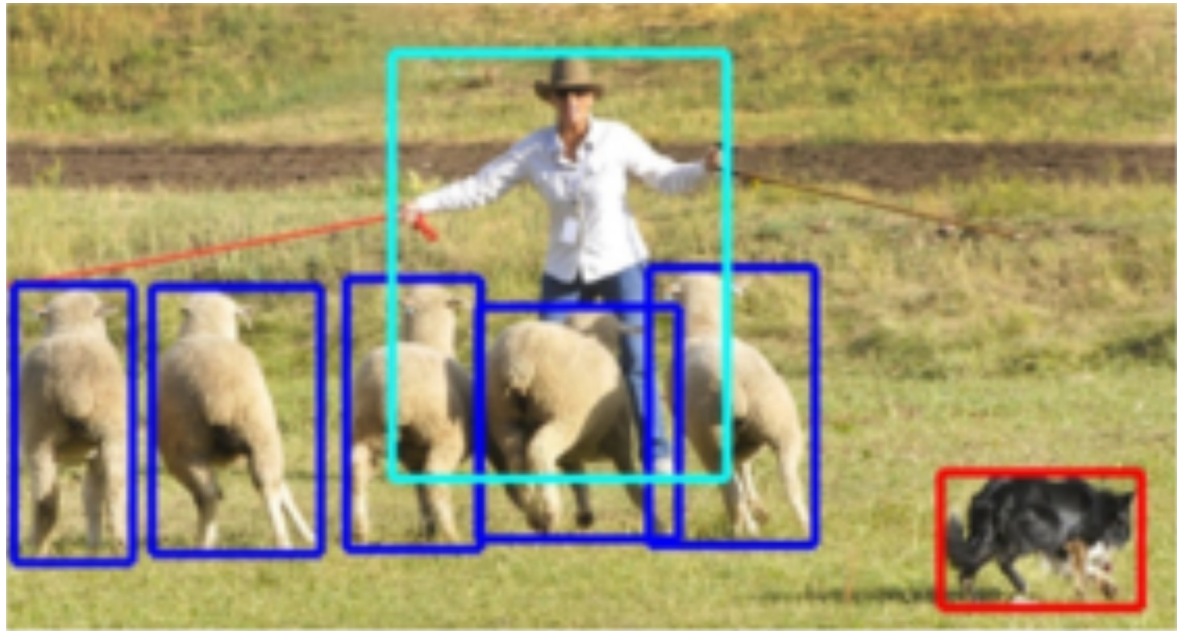
Robot navigation



Security

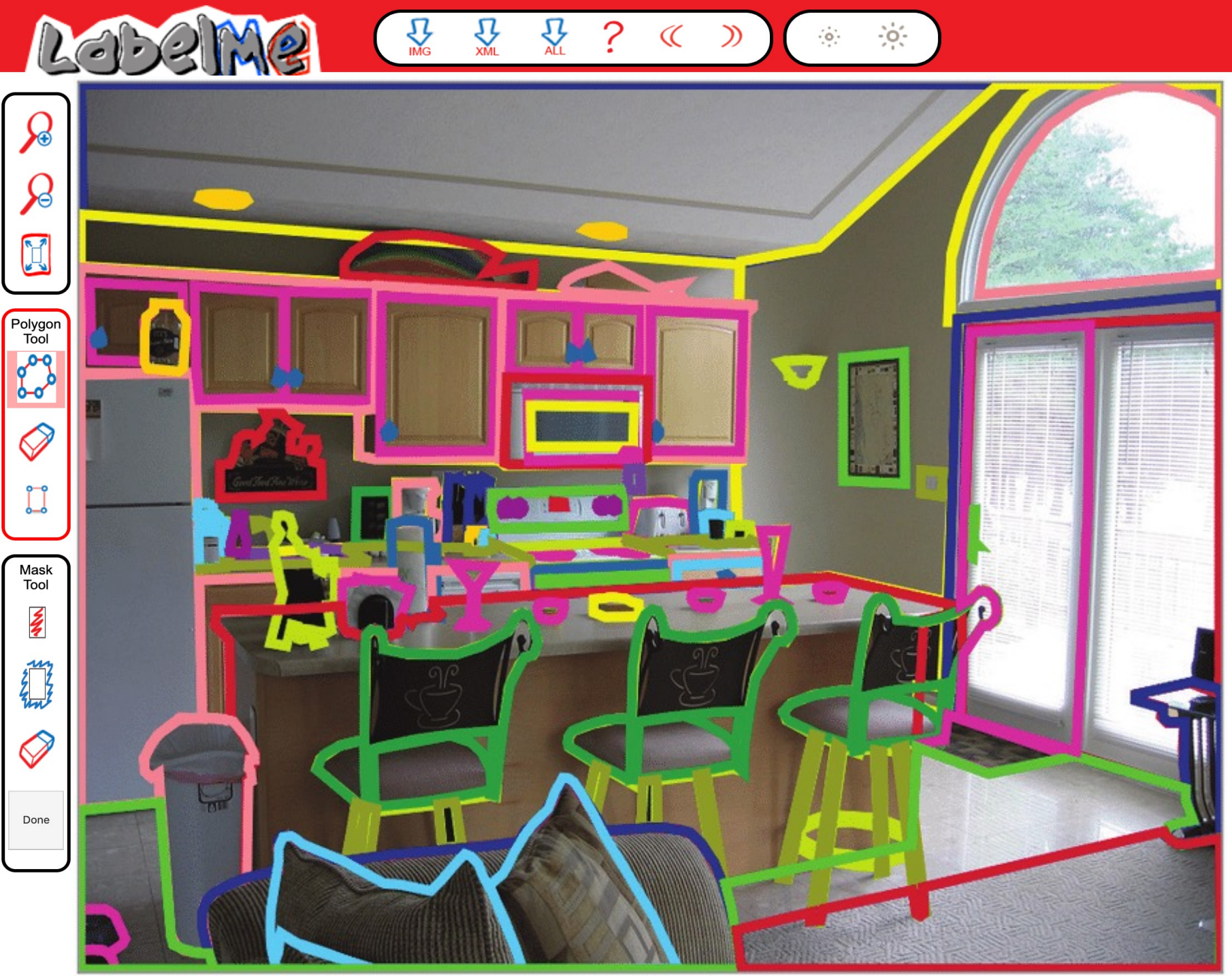


Why bounding boxes???



Object Detection

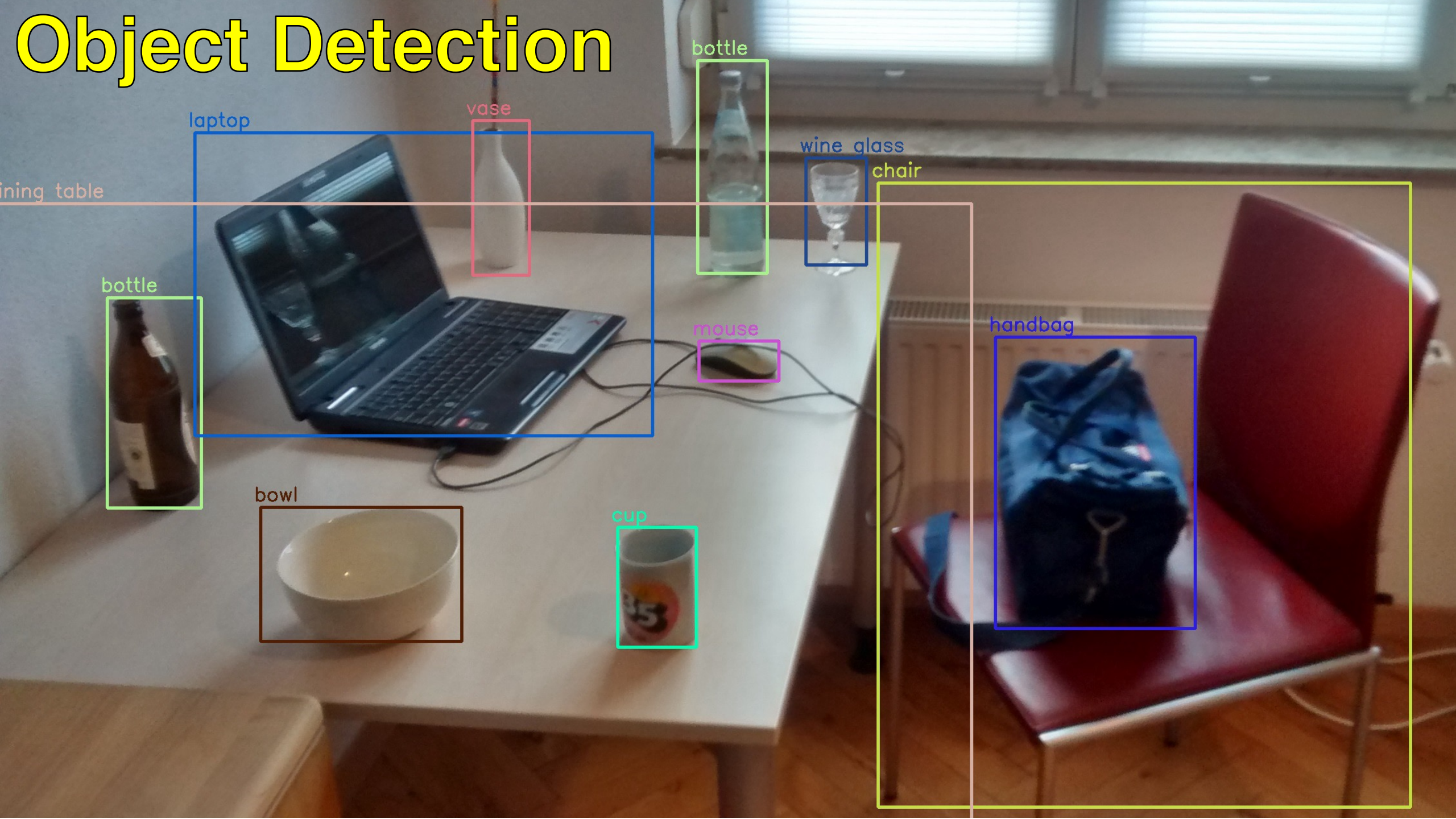
Instance segmentation
(!= semantic segmentation)

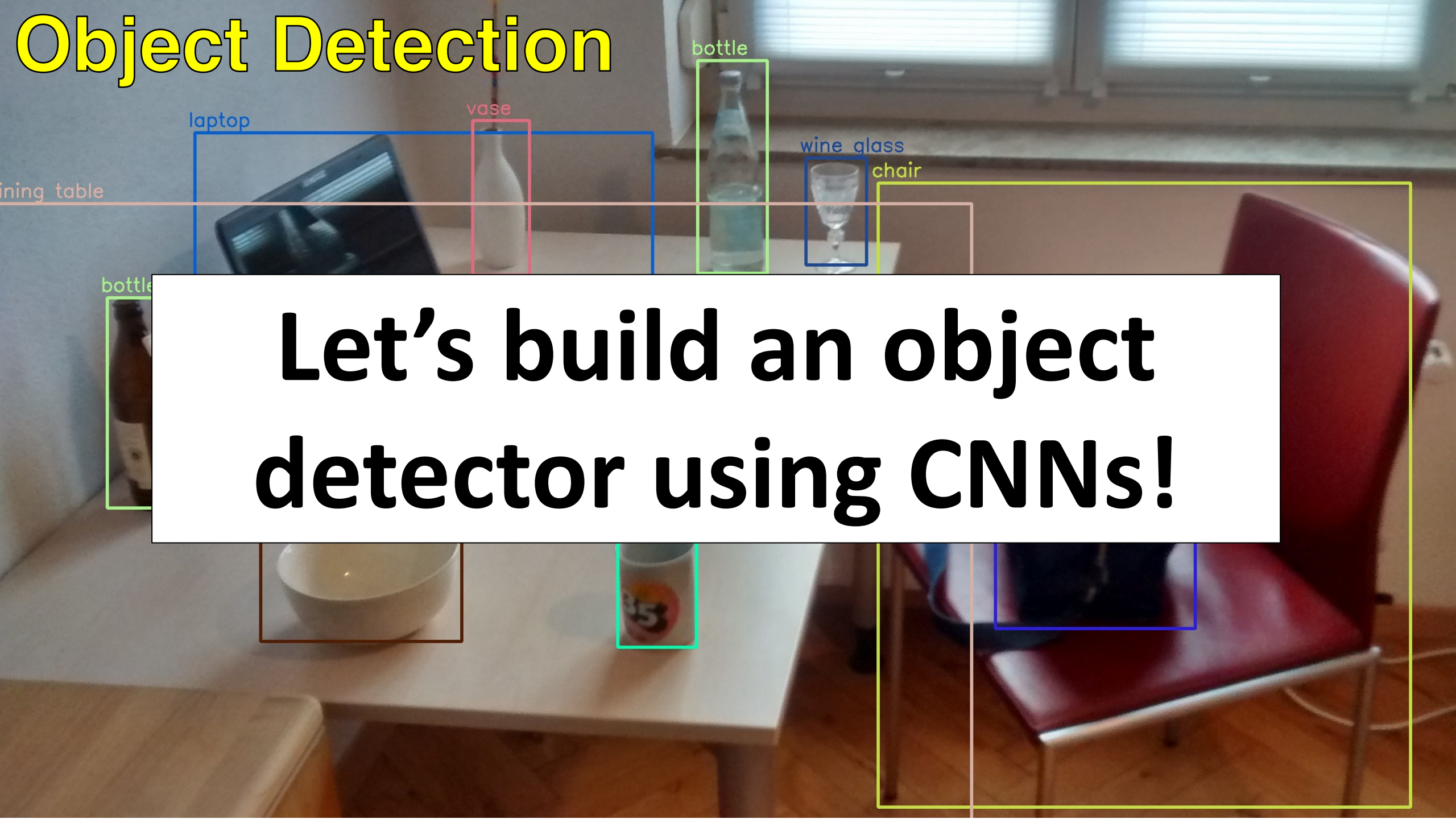


- **Training data?**
- **Expensive!!**

COCO:
80sec per object

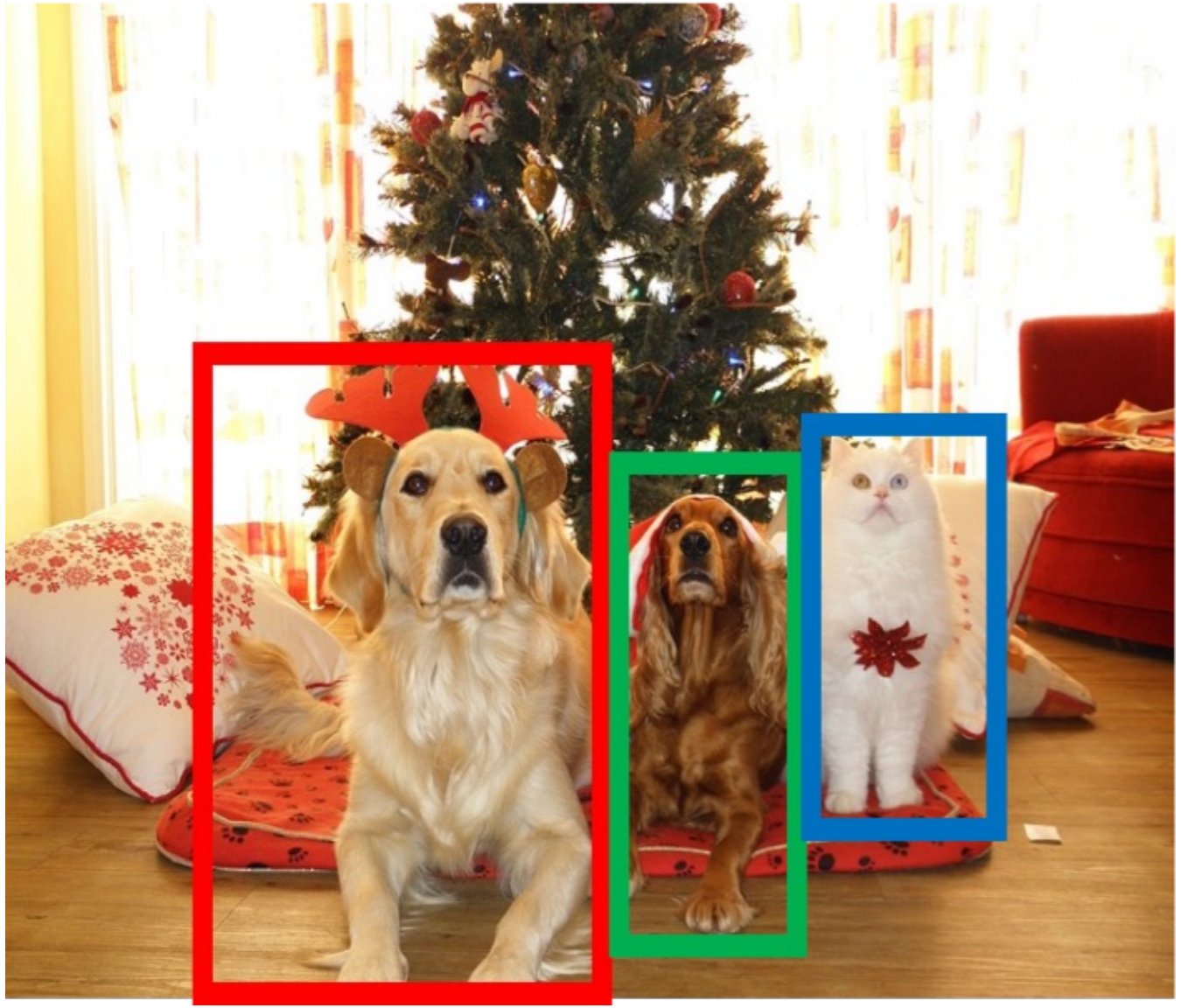
CityScapes:
1.5 hours per image


Object Detection
bottle
vase
laptop
wine glass
chair
ining table
bottle
mouse
handbag
bowl
cup

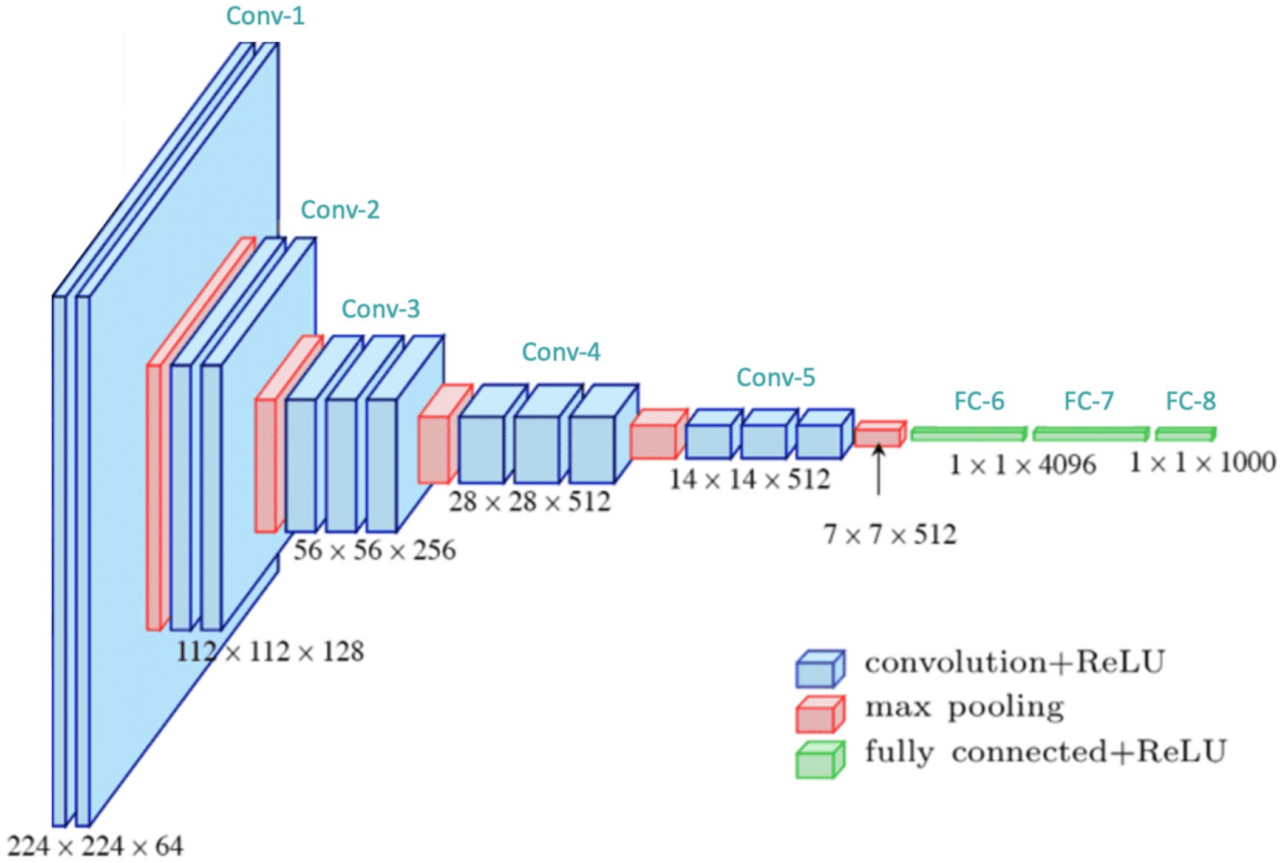

Object Detection
bottle
laptop
vase
wine glass
chair
ining table
bottle
Let’s build an object detector using CNNs!

Object Detection: Task Definition

- Input: Single RGB Image
- Output: A set of detected objects.
- For each object find:
  - Category label:
    - From a fixed known set of categories
  - Bounding box
    - Four numbers: [x, y, width, height]

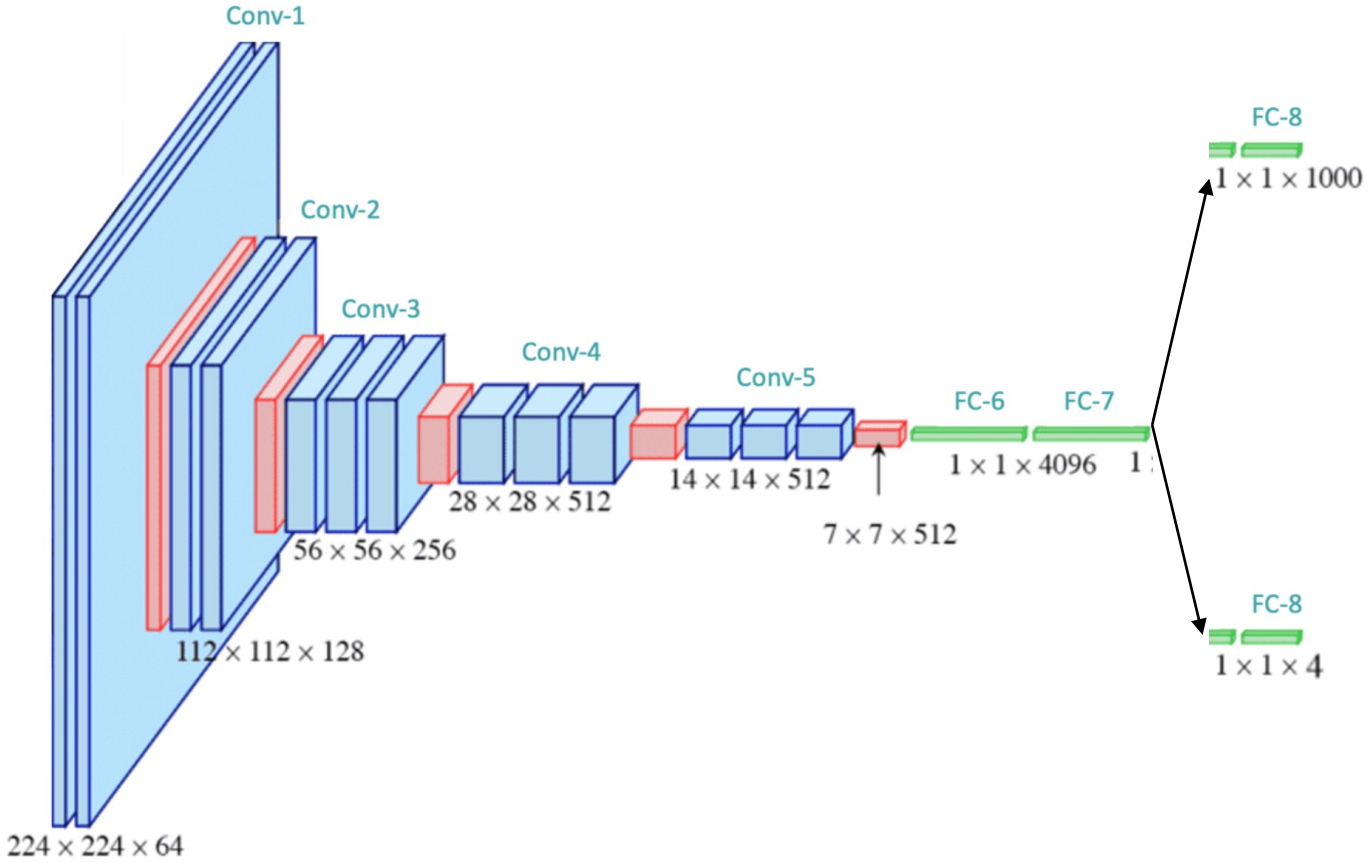


Let's start with a classification network



cat

Add a regression head (FC layer to 4 outputs)



cat: 0.9

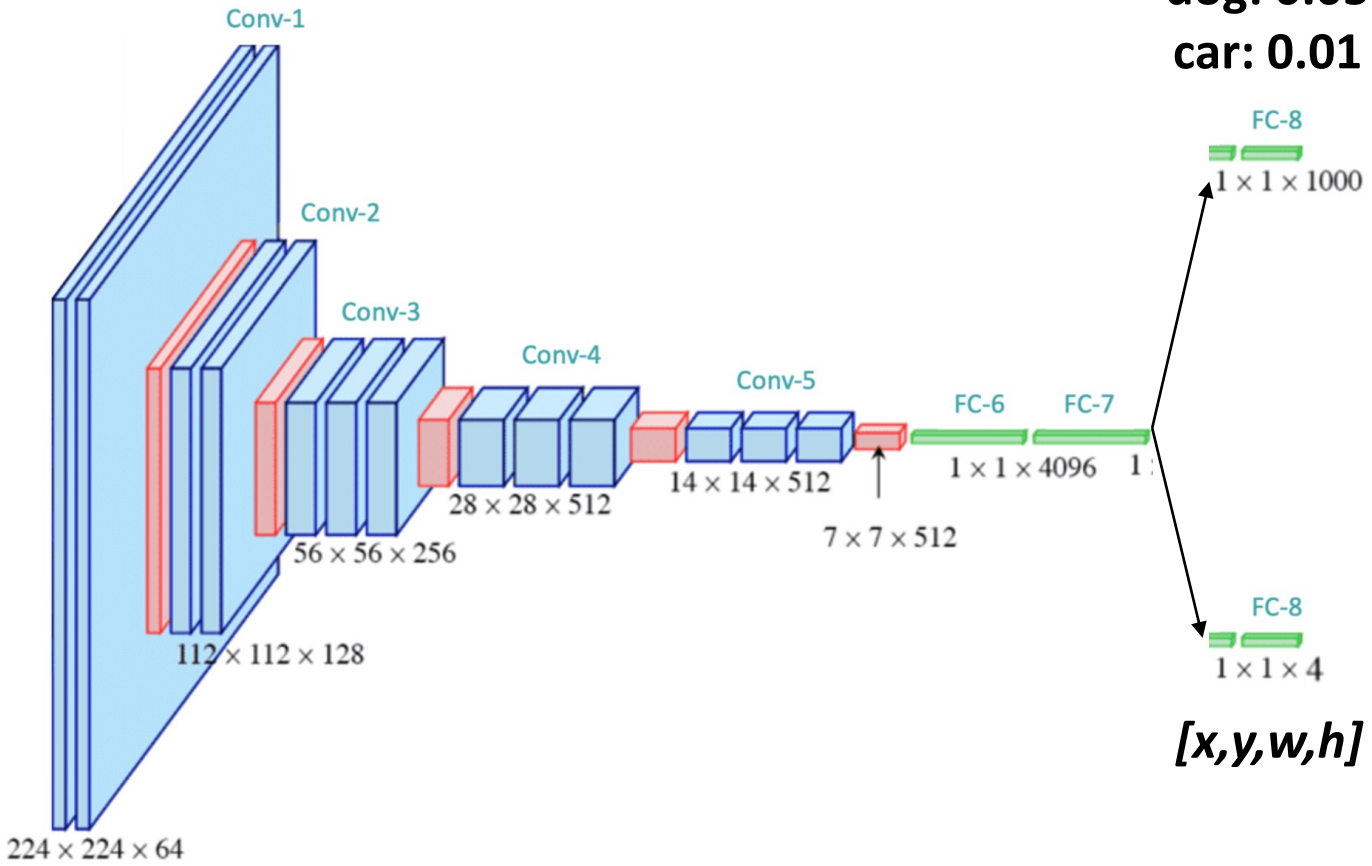
dog: 0.05

car: 0.01

… …

[x,y,w,h]

How to train this model?



cat: 0.9
dog: 0.05
car: 0.01

[x,y,w,h]

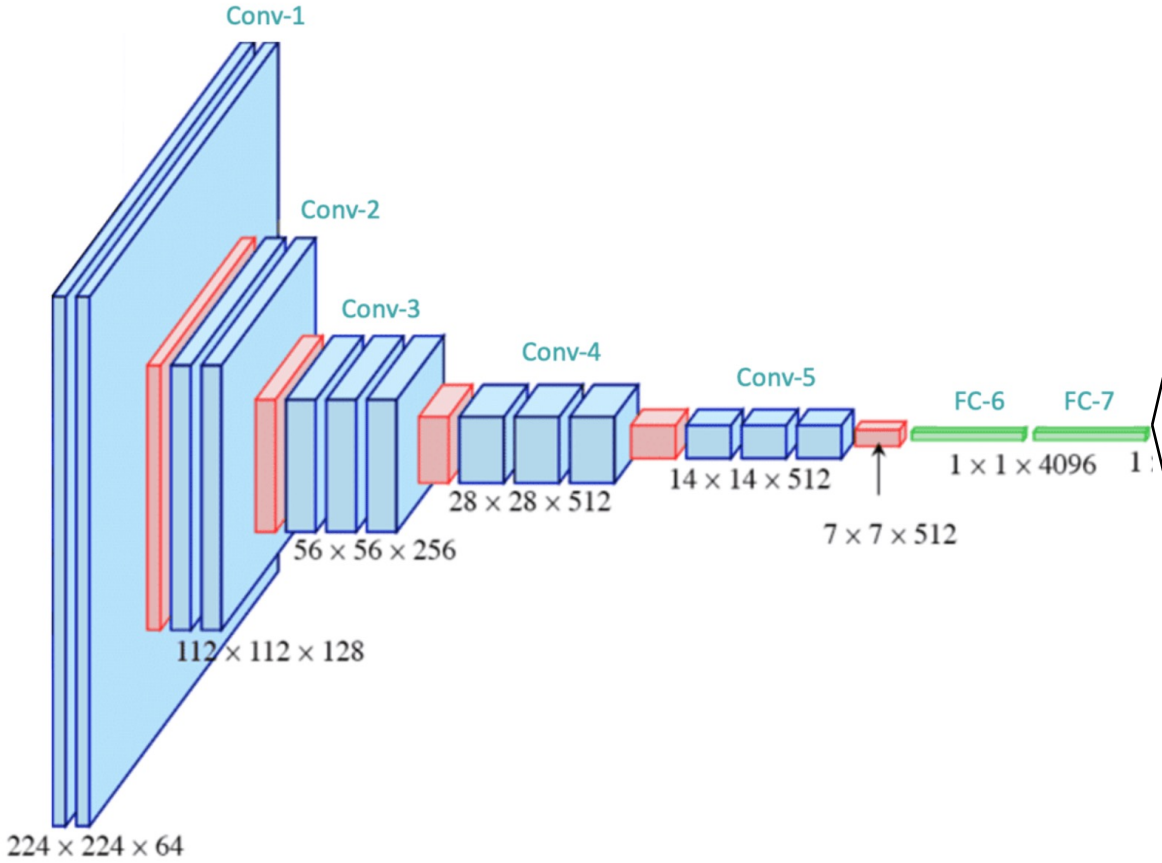
Correct label: Cat

"Loss 1" Difference

"Loss 2" Difference

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

Final loss



cat: 0.9
dog: 0.05
car: 0.01

[x,y,w,h]

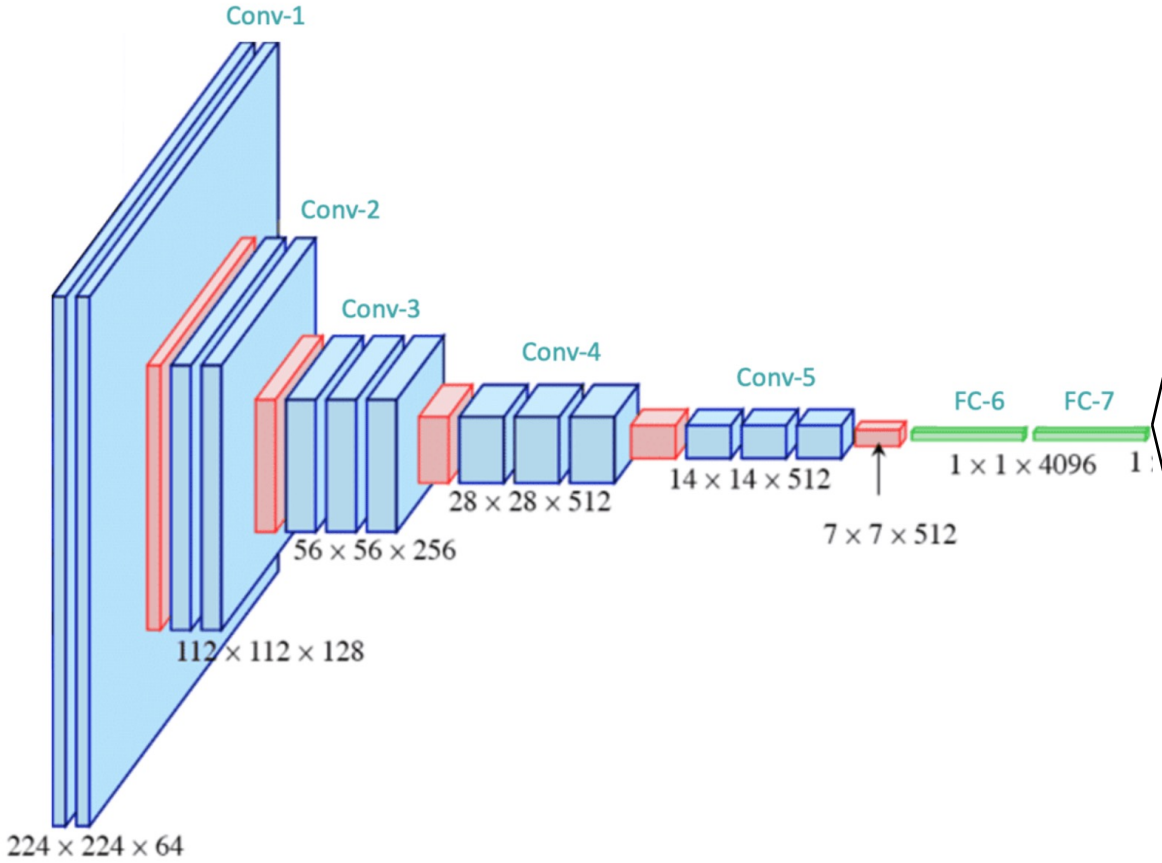
Correct label:
Cat

"Loss 1"
Difference

Loss = (L1 + L2)/2

"Loss 2"
Difference
Correct box:
(x',y',w',h')

BCE and MSE?



cat: 0.9
dog: 0.05
car: 0.01

[x,y,w,h]

Correct label:
Cat

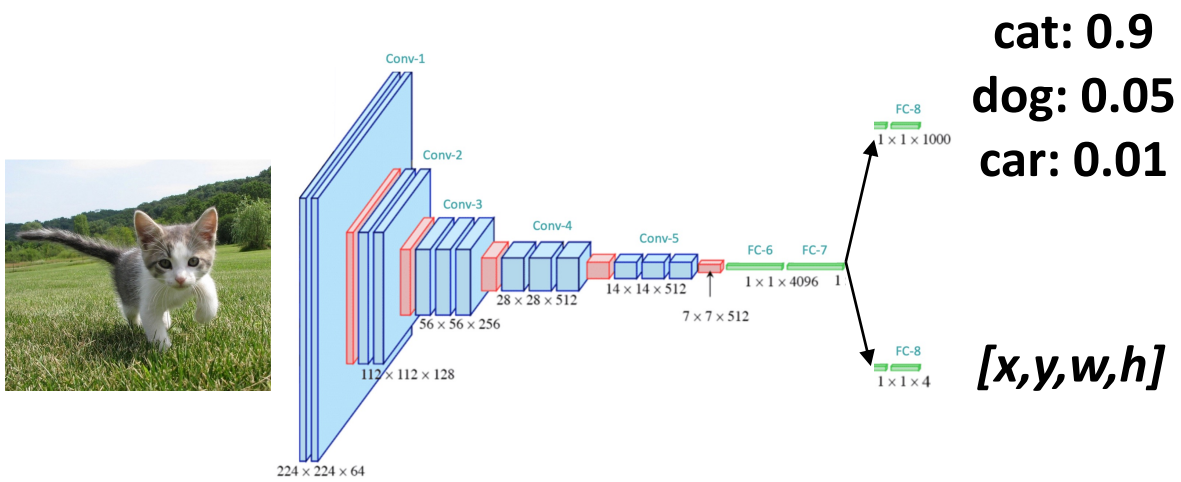
"Loss 1"
Cross-entropy

Loss = (L1 + L2)/2

"Loss 2"
Euclidean distance

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

Quiz Time: Object detection as box regression



Discuss with your neighbor (2min):

(a) Could a system like that work? If so, When?

(b) Are there any limitations?

Quiz Time: Object detection as box regression

Quiz Time: Object detection as box regression

cat: 0.9
dog: 0.9
car: 0.01

[x1,y1,w1,h1]
[x2,y2,w2,h2]

Quiz Time: Object detection as box regression

cat: 0.9
dog: 0.9
car: 0.01

[x1,y1,w1,h1]
[x2,y2,w2,h2]

Multi-label classification
[softmax to sigmoids and BCE to CE]

Need variable sized output!!

Quiz Time: Object detection as box regression

DOG, (x, y, w, h)

CAT, (x, y, w, h)

CAT, (x, y, w, h)

DUCK (x, y, w, h)

= 16 numbers

Quiz Time: Object detection as box regression

DOG, (x, y, w, h)

CAT, (x, y, w, h)

CAT, (x, y, w, h)

DUCK (x, y, w, h)

= 16 numbers

DOG, (x, y, w, h)

CAT, (x, y, w, h)

= 8 numbers

Quiz Time: Object detection as box regression

DOG, (x, y, w, h)

CAT, (x, y, w, h)

CAT, (x, y, w, h)

DUCK (x, y, w, h)

= 16 numbers

DOG, (x, y, w, h)

CAT, (x, y, w, h)

= 8 numbers

CAT, (x, y, w, h)

CAT, (x, y, w, h)

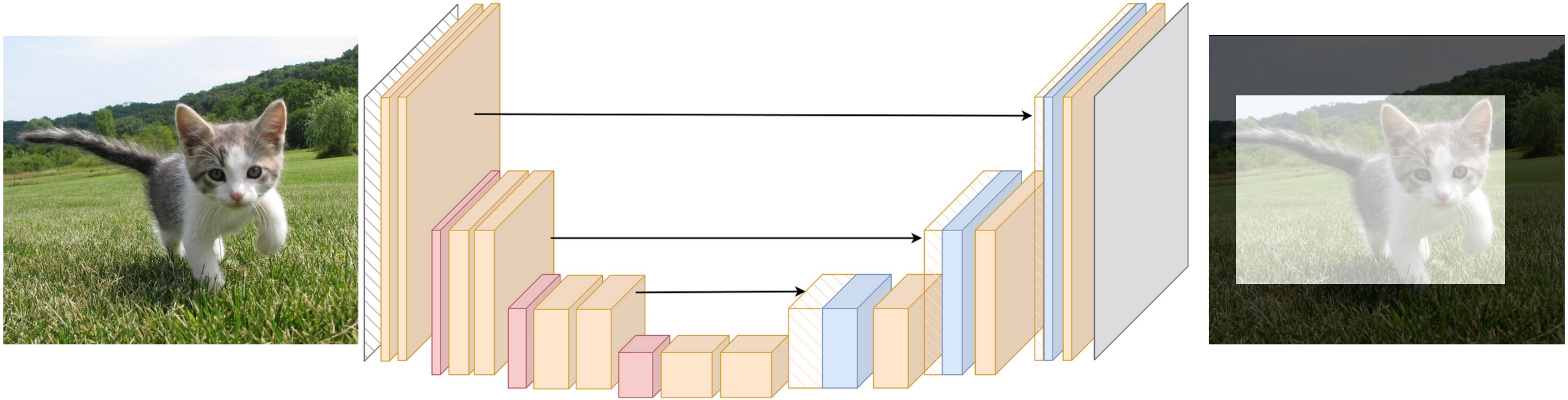
....

CAT (x, y, w, h)

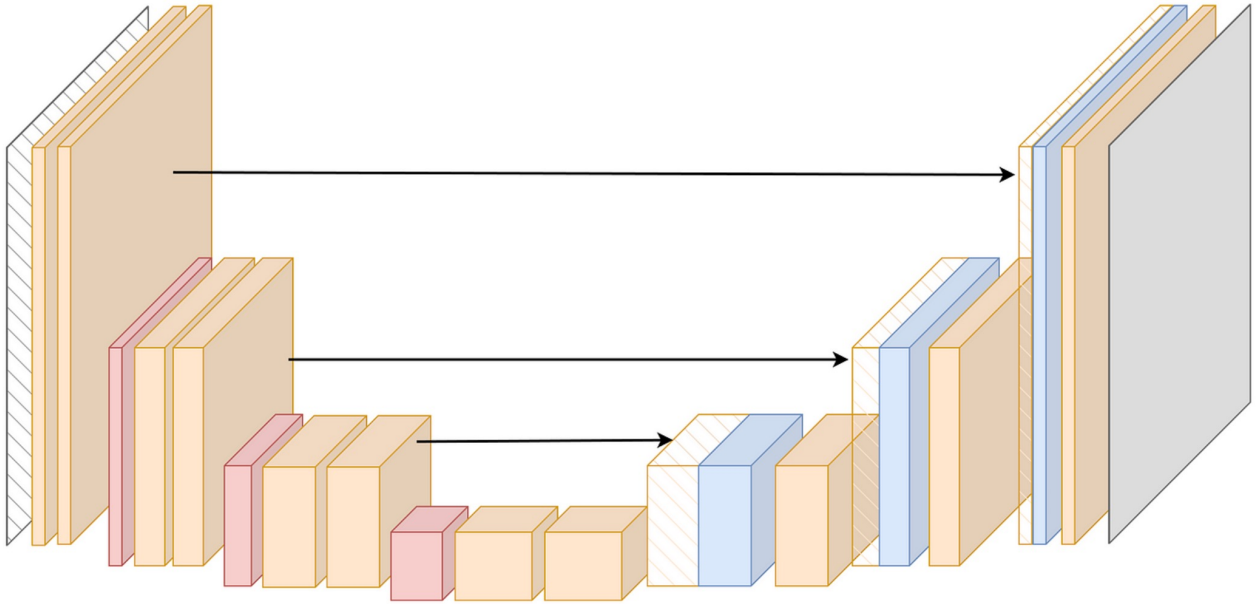
= many numbers

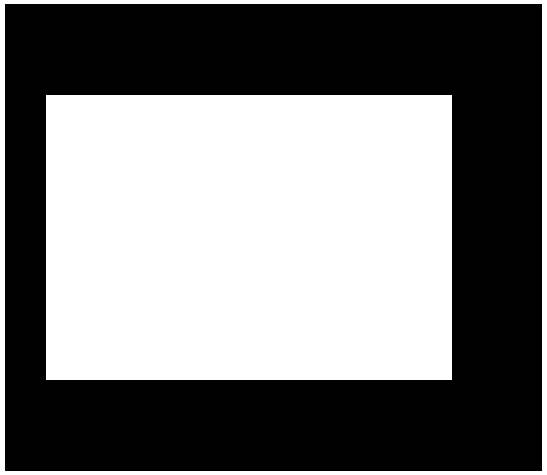
Need variable sized outputs!

I know how to classify pixels

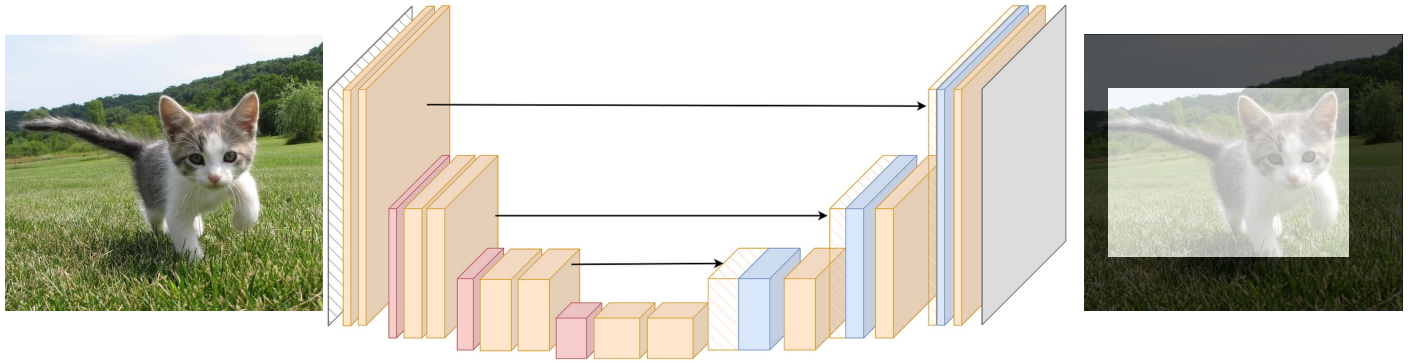


I know how to classify pixels





Quiz Time: Object detection as pixel classification



Discuss with your neighbor (2min):

(a) Could a system like that work? If so, When?

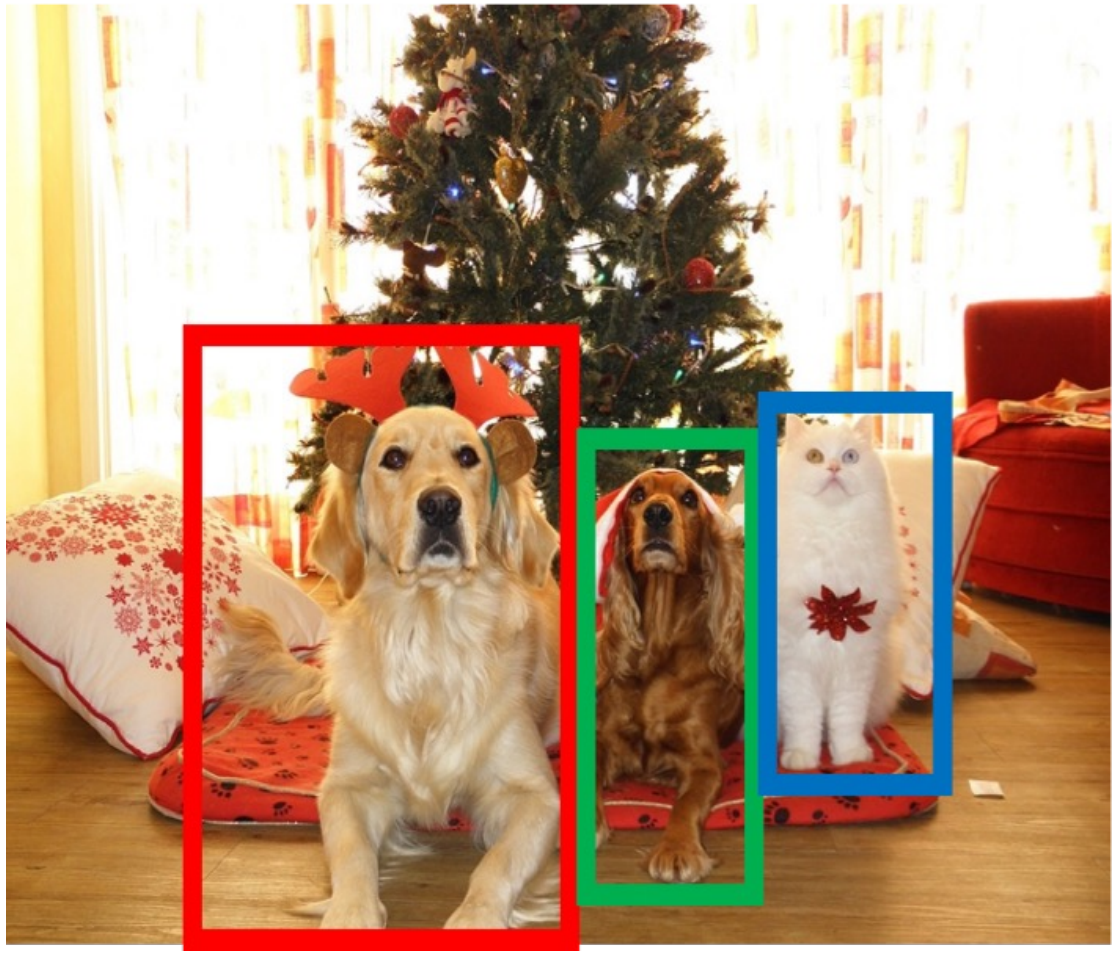
(b) Are there any limitations?

Quiz Time: Object detection as pixel classification

Quiz Time: Object detection as pixel classification

Object Detection: Task Definition

- Multiple outputs:
  - Variable number of objects per image
- Multiple types of output:
  - “ what”: category label
  - “where”: location (bounding box)
- Large images:
  - Classification works at 224x224
  - Need high resolution for detection



Today – Object Detection

- Why object detection?
- Problem Formulations and General Strategies
- The History of Object Detection (2001 – 2015)
- Object proposals and R-CNN
- **Next days:** Fast R-CNN, Faster R-CNN, comparing Boxes and evaluating object detectors, Single-stage detectors (e.g. YOLO), state-of-the-art object detectors

Today – Object Detection

- Why object detection?
- Problem Formulations and General Strategies
- **The History of Object Detection (2001 – 2015)**
- R-CNN, Fast R-CNN, Faster R-CNN
- Comparing Boxes and Evaluating Object Detectors


WHERE'S
WALDO?


Keep scanning until you find waldo

The “Waldo” model

The “Waldo” model

The “Waldo” model

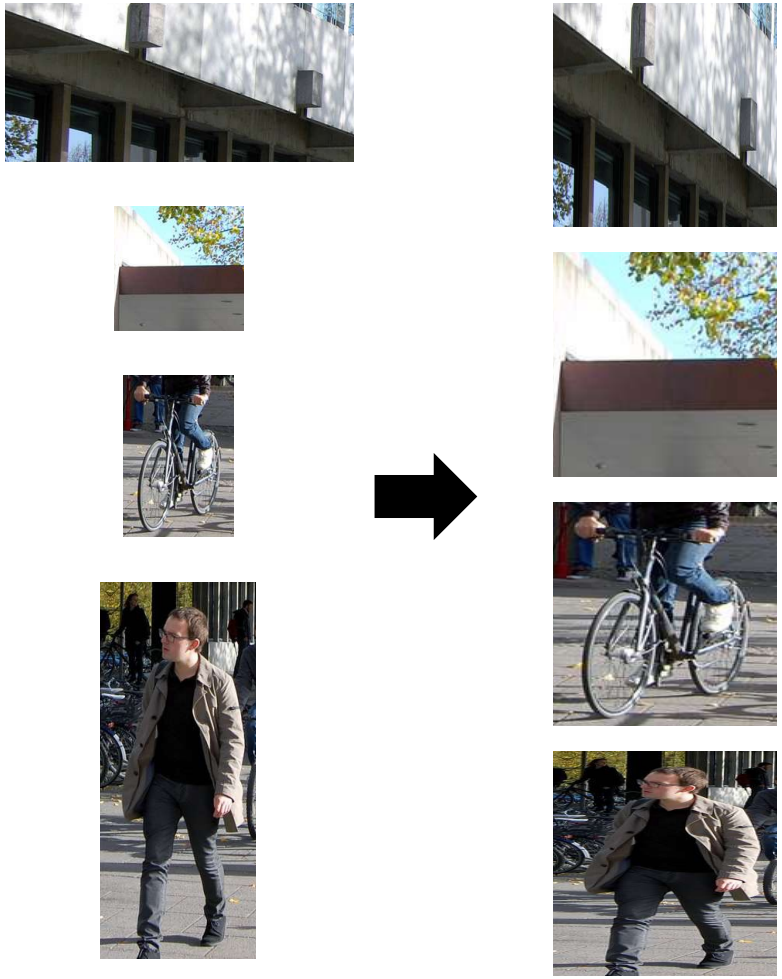
The “Waldo” model

The “Waldo” model

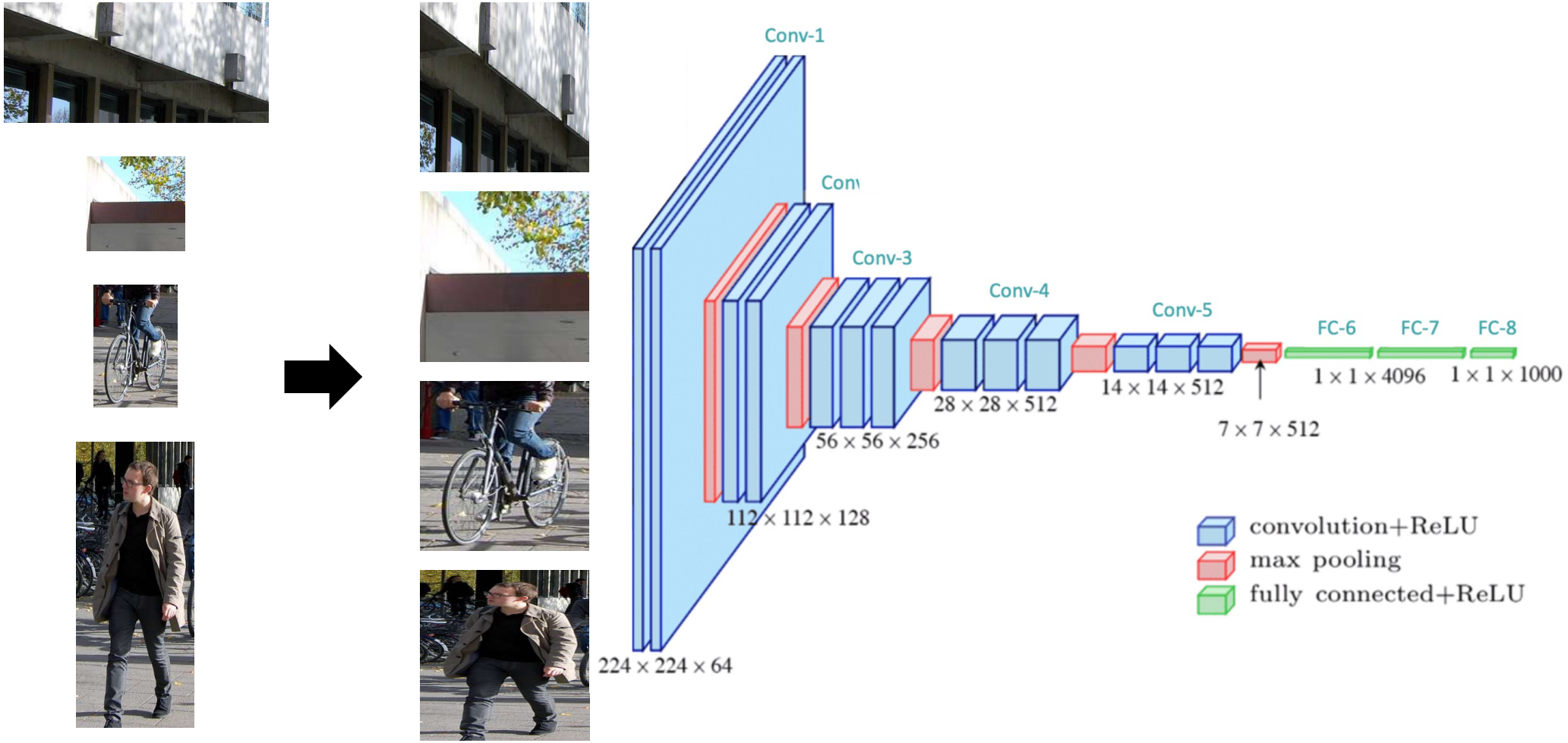
The “Waldo” model

The “Waldo” model

The “Waldo” model



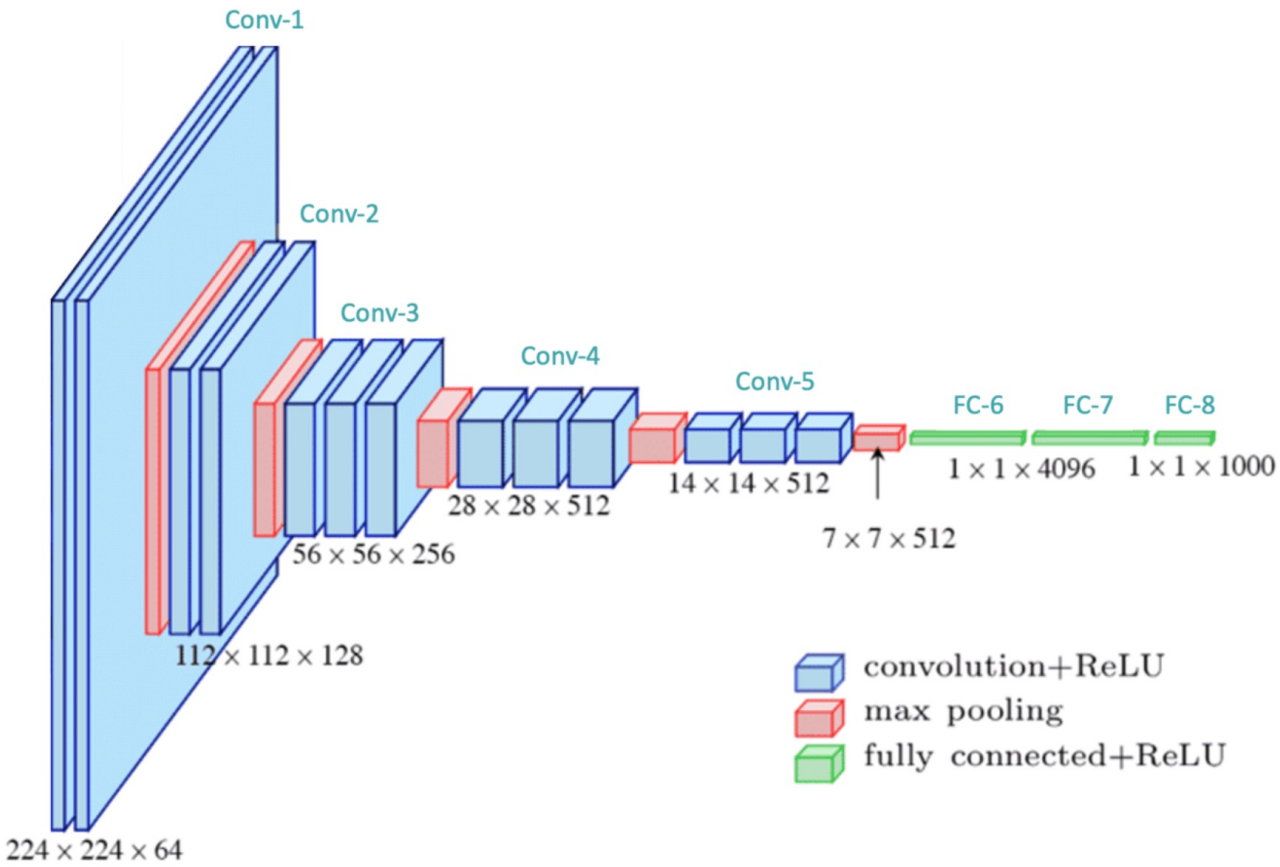
The “Waldo” model



N outputs

person: 0.9
car: 0.02
tree: 0.01
dog: 0.01
…
…

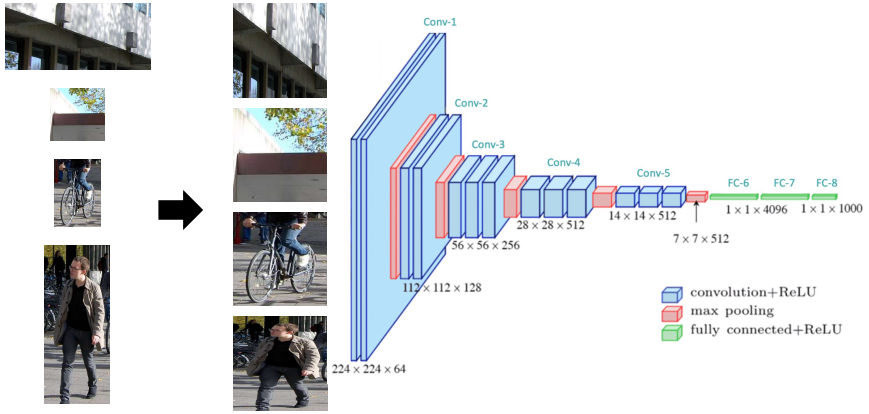
The “Waldo” model

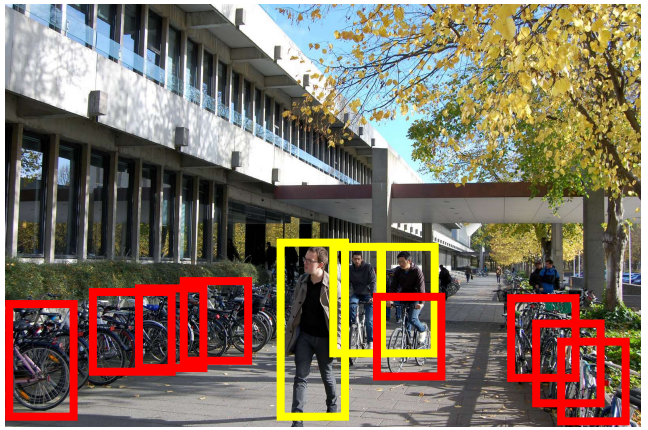


N+1 outputs

person: 0.9
car: 0.02
tree: 0.01
dog: 0.01
…
…
background: 0.01

Quiz Time: Sliding window model





Discuss with your neighbor (2min):

(a) Could a system like that work? If so, When?

(b) Are there any limitations?

Quiz Time: Sliding window model

DOG, (x, y, w, h)

CAT, (x, y, w, h)

CAT, (x, y, w, h)

DUCK (x, y, w, h)

= 16 numbers

DOG, (x, y, w, h)

CAT, (x, y, w, h)

= 8 numbers

CAT, (x, y, w, h)

CAT, (x, y, w, h)

....

CAT (x, y, w, h)

= many numbers

Sliding window → computationally prohibited

2001

Accepted Conference on Computer Vision and Pattern Recognition 2001

Rapid Object Detection using a Boosted Cascade of Simple Features

Paul Viola
viola@merl.com
Mitsubishi Electric Research Labs
201 Broadway, 8th FL
Cambridge, MA 02139

Michael Jones
mjones@crl.dec.com
Compaq CRL
One Cambridge Center
Cambridge, MA 02142

Abstract

This paper describes a machine learning approach for visual object detection which is capable of processing images extremely rapidly and achieving high detection rates. This work is distinguished by three key contributions. The first is the introduction of a new image representation called the "Integral Image" which allows the features used by our detector to be computed very quickly. The second is a learning algorithm, based on AdaBoost, which selects a small number of critical visual features from a larger set and yields extremely efficient classifiers[6]. The third contribution is a method for combining increasingly more complex classifiers in a "cascade" which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions. The cascade can be viewed as an object specific focus-of-attention mechanism which unlike previous approaches provides statistical guarantees that discarded regions are unlikely to contain the object of interest. In the domain of face detection the system yields detection rates comparable to the best previous systems. Used in real-time applications, the detector runs at 15 frames per second without resorting to image differencing or skin color detection.

1. Introduction

This paper brings together new algorithms and insights to construct a framework for robust and extremely rapid object detection. This framework is demonstrated on, and in part motivated by, the task of face detection. Toward this end we have constructed a frontal face detection system which achieves detection and false positive rates which are equivalent to the best published results [16, 12, 15, 11, 1]. This face detection system is most clearly distinguished from previous approaches in its ability to detect faces extremely rapidly. Operating on 384 by 288 pixel images, faces are detected at 15 frames per second on a conventional 700 MHz Intel Pentium III. In other face detection systems, auxiliary information, such as image differences in video sequences, or pixel color in color images, have been used to achieve high frame rates. Our system achieves high frame rates working only with the information present in a single grey scale image. These alternative sources of information can also be integrated with our system to achieve even higher frame rates.

There are three main contributions of our object detection framework. We will introduce each of these ideas briefly below and then describe them in detail in subsequent sections.

The first contribution of this paper is a new image representation called an *integral image* that allows for very fast feature evaluation. Motivated in part by the work of Papageorgiou et al. our detection system does not work directly with image intensities [10]. Like these authors we use a set of features which are reminiscent of Haar Basis functions (though we will also use related filters which are more complex than Haar filters). In order to compute these features very rapidly at many scales we introduce the integral image representation for images. The integral image can be computed from an image using a few operations per pixel. Once computed, any one of these Harr-like features can be computed at any scale or location in *constant* time.

The second contribution of this paper is a method for constructing a classifier by selecting a small number of important features using AdaBoost [6]. Within any image subwindow the total number of Harr-like features is very large, far larger than the number of pixels. In order to ensure fast classification, the learning process must exclude a large majority of the available features, and focus on a small set of critical features. Motivated by the work of Tieu and Viola, feature selection is achieved through a simple modification of the AdaBoost procedure: the weak learner is constrained so that each weak classifier returned can depend on only a

2011-2014

Object Detection with Discriminatively Trained Part Based Models

Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester and Deva Ramanan

1

Abstract—We describe an object detection system based on mixtures of multiscale deformable part models. Our system is able to represent highly variable object classes and achieves state-of-the-art results in the PASCAL object detection challenges. While deformable part models have become quite popular, their value had not been demonstrated on difficult benchmarks such as the PASCAL datasets. Our system relies on new methods for discriminative training with partially labeled data. We combine a margin-sensitive approach for data-mining hard negative examples with a formalism we call *latent SVM*. A latent SVM is a reformulation of MI-SVM in terms of latent variables. A latent SVM is semi-convex and the training problem becomes convex once latent information is specified for the positive examples. This leads to an iterative training algorithm that alternates between fixing latent values for positive examples and optimizing the latent SVM objective function.

Index Terms—Object Recognition, Deformable Models, Pictorial Structures, Discriminative Training, Latent SVM

1 INTRODUCTION

Object recognition is one of the fundamental challenges in computer vision. In this paper we consider the problem of detecting and localizing generic objects from categories such as people or cars in static images. This is a difficult problem because objects in such categories can vary greatly in appearance. Variations arise not only from changes in illumination and viewpoint, but also due to non-rigid deformations, and intraclass variability in shape and other visual properties. For example, people wear different clothes and take a variety of poses while cars come in a various shapes and colors.

We describe an object detection system that represents highly variable objects using mixtures of multiscale deformable part models. These models are trained using a discriminative procedure that only requires bounding boxes for the objects in a set of images. The resulting system is both efficient and accurate, achieving state-of-the-art results on the PASCAL VOC benchmarks [11]–[13] and the INRIA Person dataset [10].

Our approach builds on the pictorial structures framework [15], [20]. Pictorial structures represent objects by a collection of parts arranged in a deformable configuration. Each part captures local appearance properties of an object while the deformable configuration is characterized by spring-like connections between certain pairs of parts.

Deformable part models such as pictorial structures provide an elegant framework for object detection. Yet

it has been difficult to establish their value in practice. On difficult datasets deformable part models are often outperformed by simpler models such as rigid templates [10] or bag-of-features [44]. One of the goals of our work is to address this performance gap.

While deformable models can capture significant variations in appearance, a single deformable model is often not expressive enough to represent a rich object category. Consider the problem of modeling the appearance of bicycles in photographs. People build bicycles of different types (e.g., mountain bikes, tandems, and 19th-century cycles with one big wheel and a small one) and view them in various poses (e.g., frontal versus side views). The system described here uses mixture models to deal with these more significant variations.

We are ultimately interested in modeling objects using "visual grammars". Grammar based models (e.g. [16], [24], [45]) generalize deformable part models by representing objects using variable hierarchical structures. Each part in a grammar based model can be defined directly or in terms of other parts. Moreover, grammar based models allow for, and explicitly model, structural variations. These models also provide a natural framework for sharing information and computation between different object classes. For example, different m... might share reusable parts.

Although grammar based models ar... goal, we have adopted a research ... which we gradually move tow... maintaining a high level ... performance by enrich... Simple models h... ticated model... machine... ampl...

- P.F. Felzenszwalb is with the Department of Computer Science, University of Chicago. E-mail: pff@cs.uchicago.edu
- R.B. Girshick is with the Department of Computer Science, University of Chicago. E-mail: rbg@cs.uchicago.edu
- D. McAllester is with the Toyota Technological Institute at Chicago. E-mail: mcallester@tti-c.org
- D. Ramanan is with the Department of Computer Science, UC Irvine. E-mail: dramanan@ics.uci.edu

Today
Object detection

- Various Problem Formulations
- General Strategies for Object Detection
  - Single Object Localization
  - Detection as Regression
  - **Detection as Classification → R-CNN**
- Comparing Boxes
- Evaluating Object Detectors

Detection as classification

CAT? NO

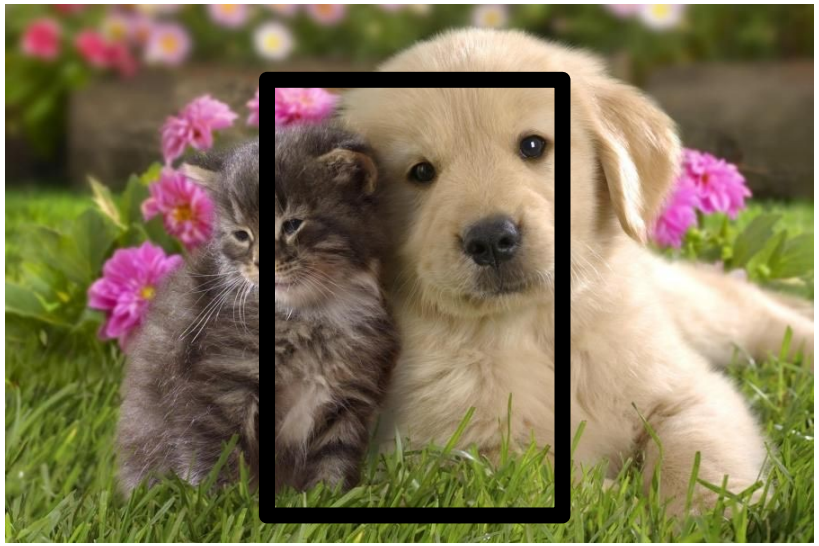
DOG? NO

Detection as classification

CAT? YES

DOG? NO

Detection as classification



CAT? NO

DOG? NO

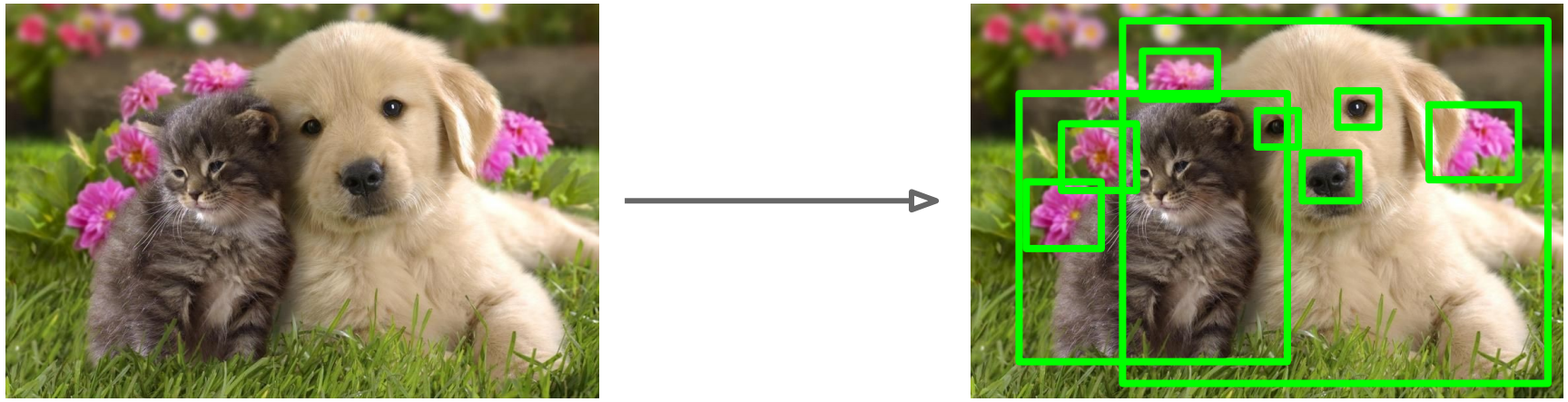
Detection as Classification

Instead of looking at all possible spatial windows at multiple positions and scales (sliding window)

Look only at a **tiny** subset of carefully selected possible positions

Region Proposals

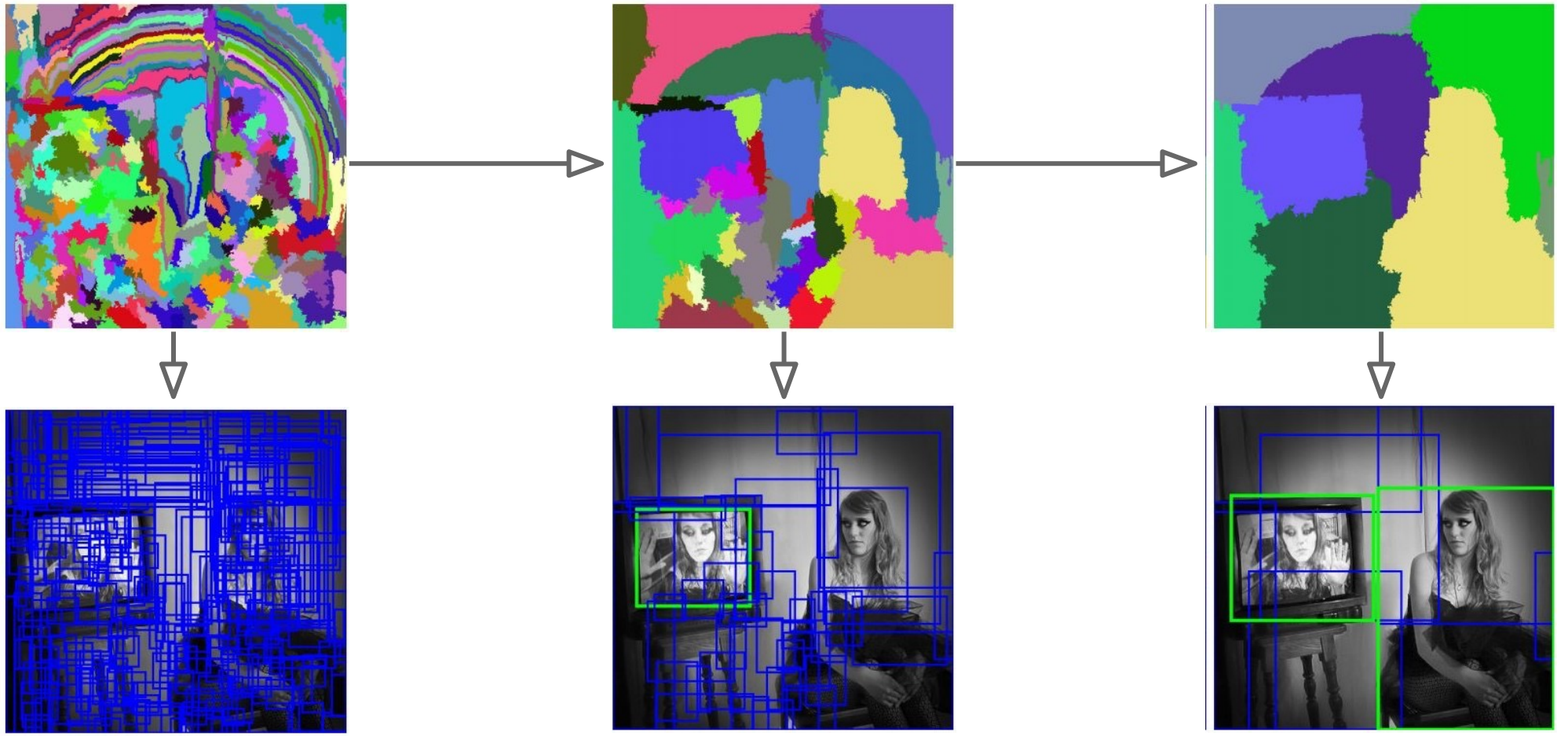
- Find “blobby” image regions that are likely to contain objects
- “Class-agnostic” object detector
- Look for “blob-like” regions



Region Proposals: Selective Search (SS)

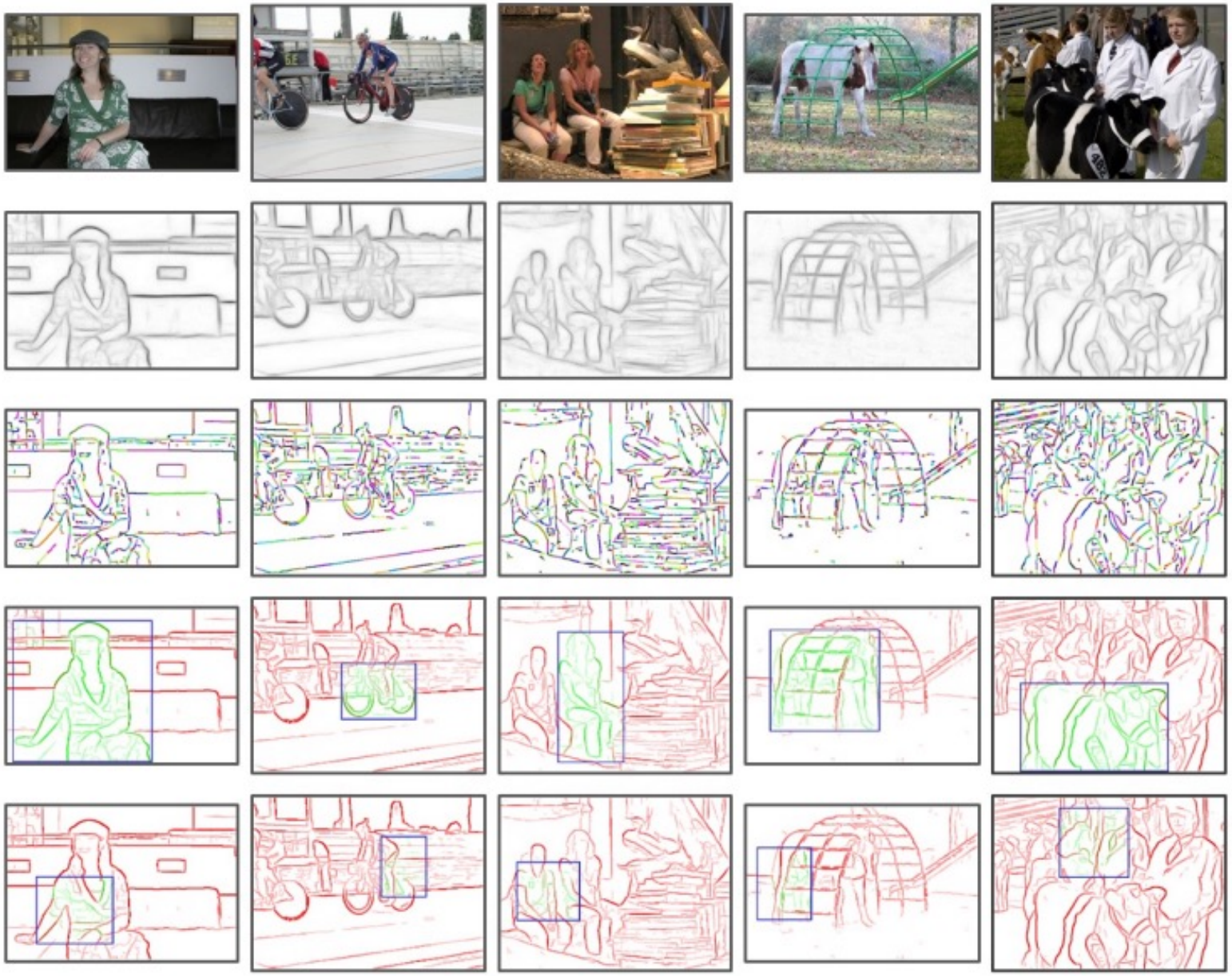
Bottom-up segmentation, merging regions at multiple scales

Convert regions to boxes



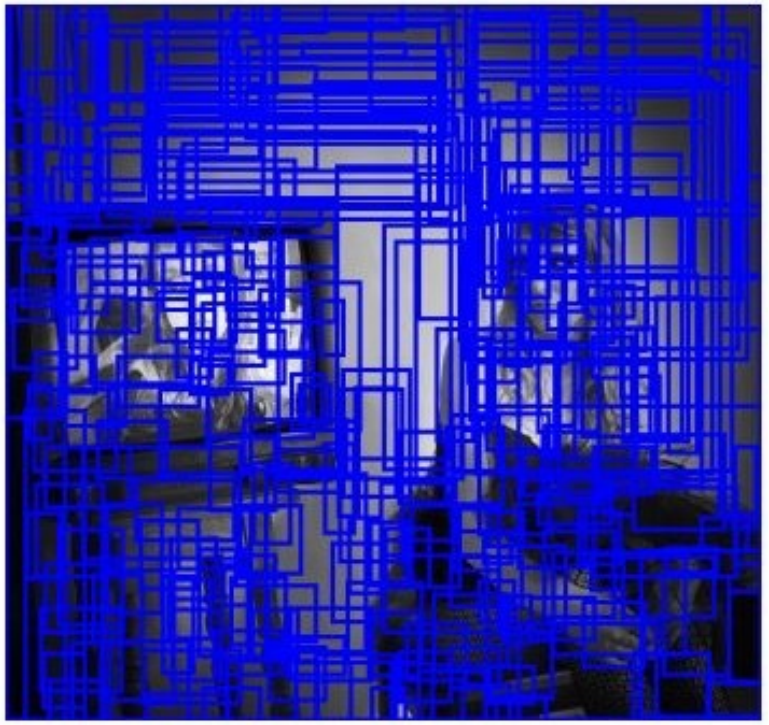
Uijlings et al, “Selective Search for Object Recognition”, IJCV 2013

Region Proposals: Edge boxes



[Zitnick and Dollar, ECCV2014]

Evaluating Object Proposals



- **High recall (all GT objects)**
- **Ideally, using as few as possible**

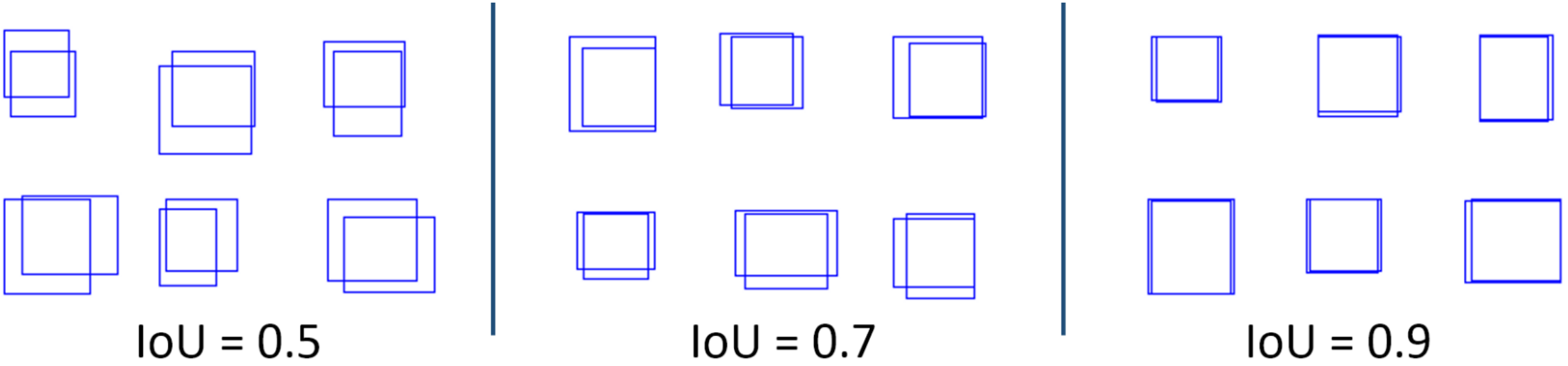
Evaluating Object Proposals

- **Recall/ Detection rate:**
  Percentage of objects with at least one “good” object proposal. “Good” → IoU> k

Uijlings et al, “Selective Search for Object Recognition”, IJCV 2013

Evaluating Object Proposals

- **Recall/ Detection rate:**
  Percentage of objects with at least one "good" object proposal. "Good" → IoU> k

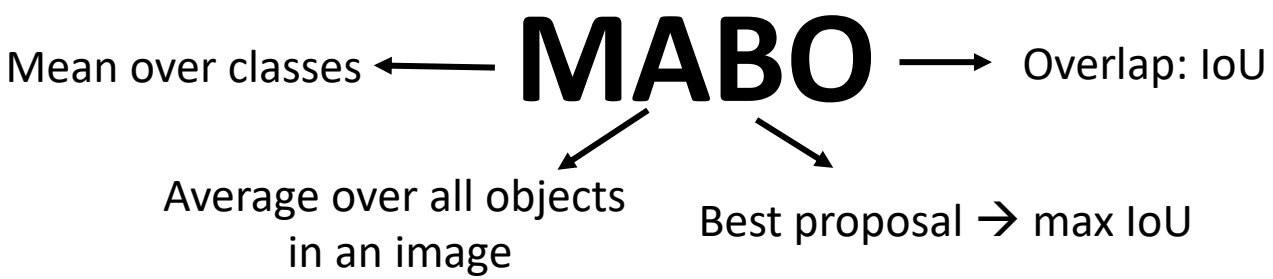


Uijlings et al, "Selective Search for Object Recognition", IJCV 2013

Evaluating Object Proposals

- **Recall/ Detection rate:**
  Percentage of objects with at least one "good" object proposal. "Good" → IoU> k

- **MABO (Mean Average Best Overlap)**

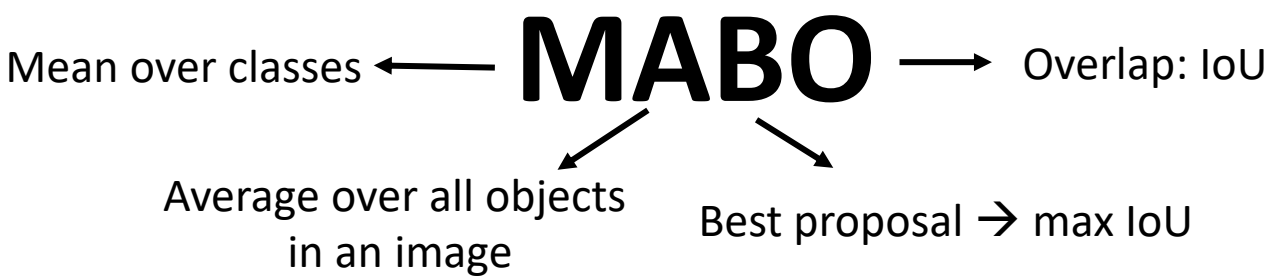


Uijlings et al, "Selective Search for Object Recognition", IJCV 2013

Evaluating Object Proposals

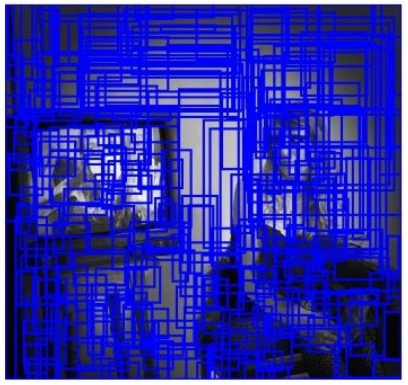
- **Recall/ Detection rate:**
  Percentage of objects with at least one "good" object proposal. "Good" → IoU> k

- **MABO (Mean Average Best Overlap)**



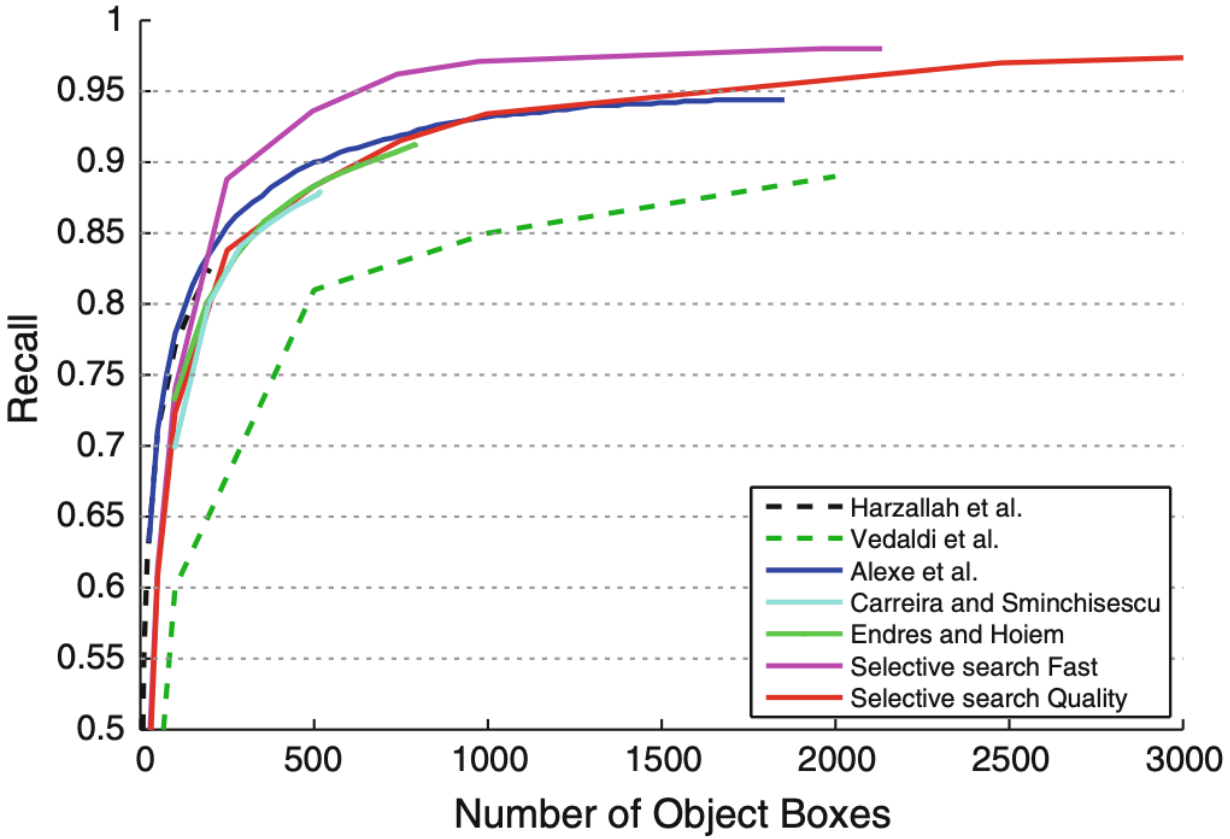
$$\text{ABO} = \frac{1}{|G^c|} \sum_{g_i^c \in G^c} \max_{l_j \in L} \text{Overlap}(g_i^c, l_j)$$

$$\text{Overlap}(g_i^c, l_j) = \frac{\text{area}(g_i^c) \cap \text{area}(\text{l}_\text{j})}{\text{area}(g_i^c) \cup \text{area}(\text{l}_\text{j})}$$

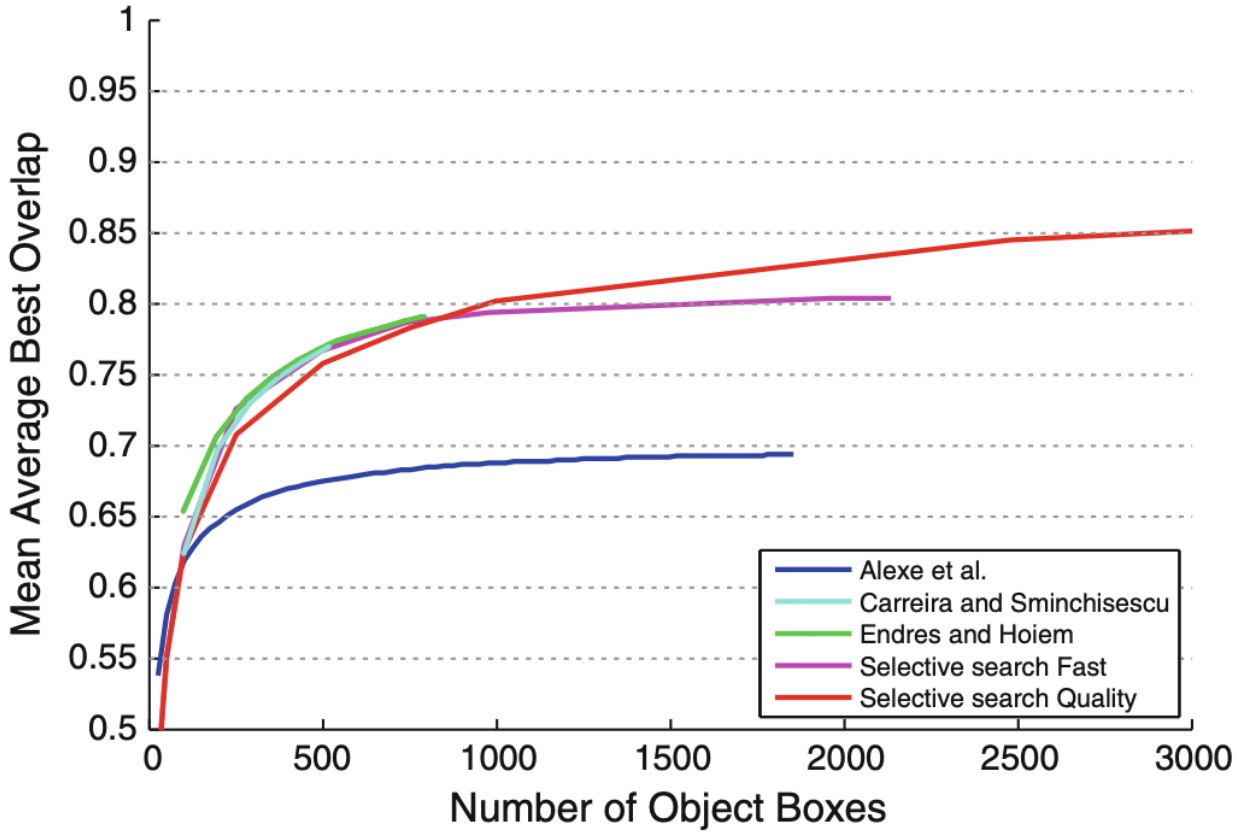


Uijlings et al, "Selective Search for Object Recognition", IJCV 2013

Evaluating Object Proposals (SS)



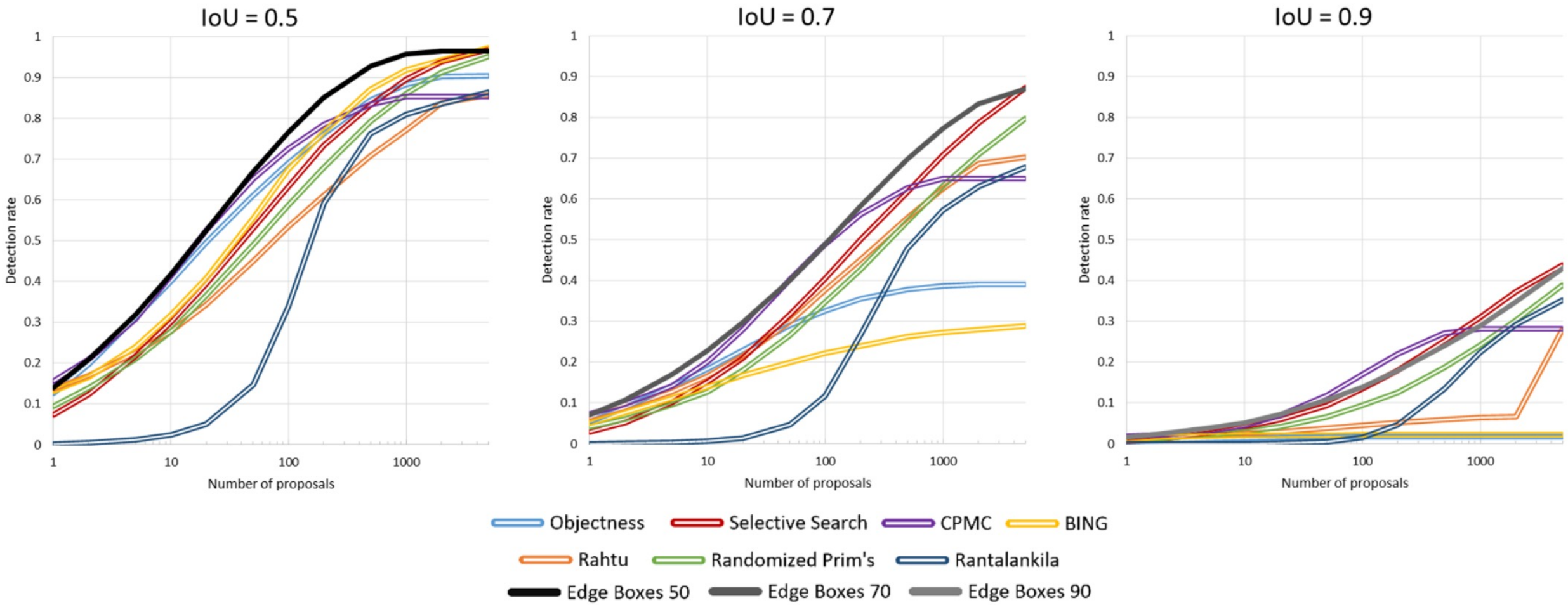
(a) Trade-off between number of object locations and the Pascal Recall criterion.



(b) Trade-off between number of object locations and the MABO score.

Uijlings et al, "Selective Search for Object Recognition", IJCV 2013

Evaluating Object Proposals (EdgeBoxes)



Uijlings et al, "Selective Search for Object Recognition", IJCV 2013

Rich feature hierarchies for accurate object detection and semantic segmentation

Ross Girshick[1] Jeff Donahue[1,2] Trevor Darrell[1,2] Jitendra Malik[1]
[1]UC Berkeley and [2]ICSI
{rbg,jdonahue,trevor,malik}@eecs.berkeley.edu

Abstract

Object detection performance, as measured on the canonical PASCAL VOC dataset, has plateaued in the last few years. The best-performing methods are complex ensemble systems that typically combine multiple low-level image features with high-level context. In this paper, we propose a simple and scalable detection algorithm that improves mean average precision (mAP) by more than 30% relative to the previous best result on VOC 2012—achieving a mAP of 53.3%. Our approach combines two key insights: (1) one can apply high-capacity convolutional neural networks (CNNs) to bottom-up region proposals in order to localize and segment objects and (2) when labeled training data is scarce, supervised pre-training for an auxiliary task, followed by domain-specific fine-tuning, yields a significant performance boost. Since we combine region proposals with CNNs, we call our method R-CNN: *Regions with CNN features. We also present experiments that provide insight into what the network learns, revealing a rich hierarchy of image features. Source code for the complete system is available at* http://www.cs.berkeley.edu/~rbg/rcnn.

1. Introduction

Features matter. The last decade of progress on various visual recognition tasks has been based considerably on the use of SIFT [26] and HOG [7]. But if we look at performance on the canonical visual recognition task, PASCAL VOC object detection [12], it is generally acknowledged that progress has been slow during 2010-2012, with small gains obtained by building ensemble systems and employing minor variants of successful methods.

SIFT and HOG are blockwise orientation histograms, a representation we could associate roughly with complex cells in V1, the first cortical area in the primate visual pathway. But we also know that recognition occurs several stages downstream, which suggests that there might be hierarchical, multi-stage processes for computing features that are even more informative for visual recognition.

Fukushima's "neocognitron" [16], a biologically-inspired hierarchical and shift-invariant model for pattern recognition, was an early attempt at just such a process. The neocognitron, however, lacked a supervised training algorithm. LeCun *et al*. [23] provided the missing algorithm by showing that stochastic gradient descent, via backpropagation, can train convolutional neural networks (CNNs), a class of models that extend the neocognitron.

CNNs saw heavy use in the 1990s (*e.g.*, [24]), but then fell out of fashion, particularly in computer vision, with the rise of support vector machines. In 2012, Krizhevsky *et al*. [22] rekindled interest in CNNs by showing substantially higher image classification accuracy on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [9, 10]. Their success resulted from training a large CNN on 1.2 million labeled images, together with a few twists on LeCun's CNN (*e.g.*, $\max(x, 0)$ rectifying non-linearities and "dropout" regularization).

The significance of the ImageNet result was vigorously debated during the ILSVRC 2012 workshop. The central issue can be distilled to the following: To what extent do the CNN classification results on ImageNet generalize to object detection results on the PASCAL VOC Challenge?

We answer this question decisively by bridging the chasm between image classification and object detection. This paper is the first to show that a CNN can lead to dra-

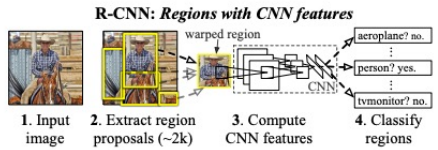
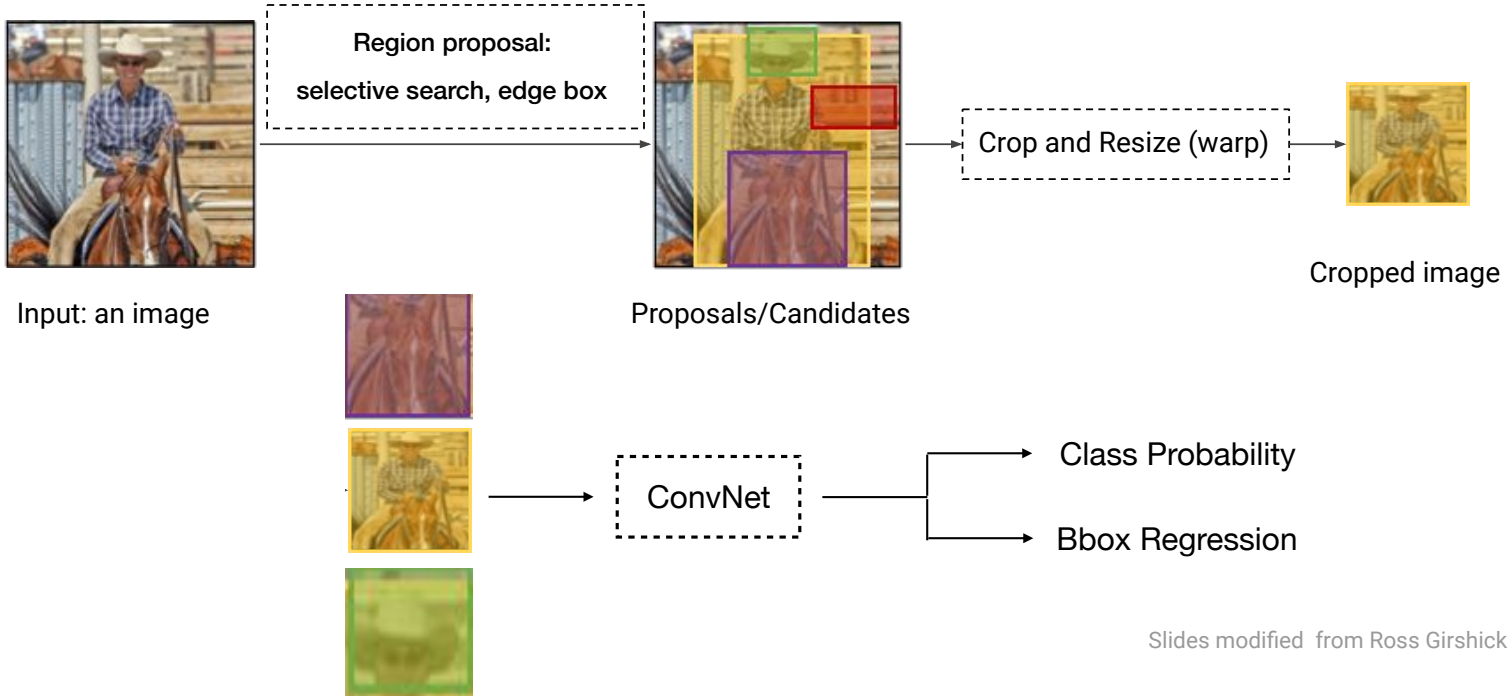


Figure 1: Object detection system overview. Our system (1) takes an input image, (2) extracts around 2000 bottom-up region proposals, (3) computes features for each proposal using a large convolutional neural network (CNN), and then (4) classifies each region using class-specific linear SVMs. R-CNN achieves a mean average precision (mAP) of **53.7% on PASCAL VOC 2010**. For comparison, [32] reports 35.1% mAP using the same region proposals, but with a spatial pyramid and bag-of-visual-words approach. The popular deformable part models perform at 33.4%.

All together:
R-CNN: Region-based CNN

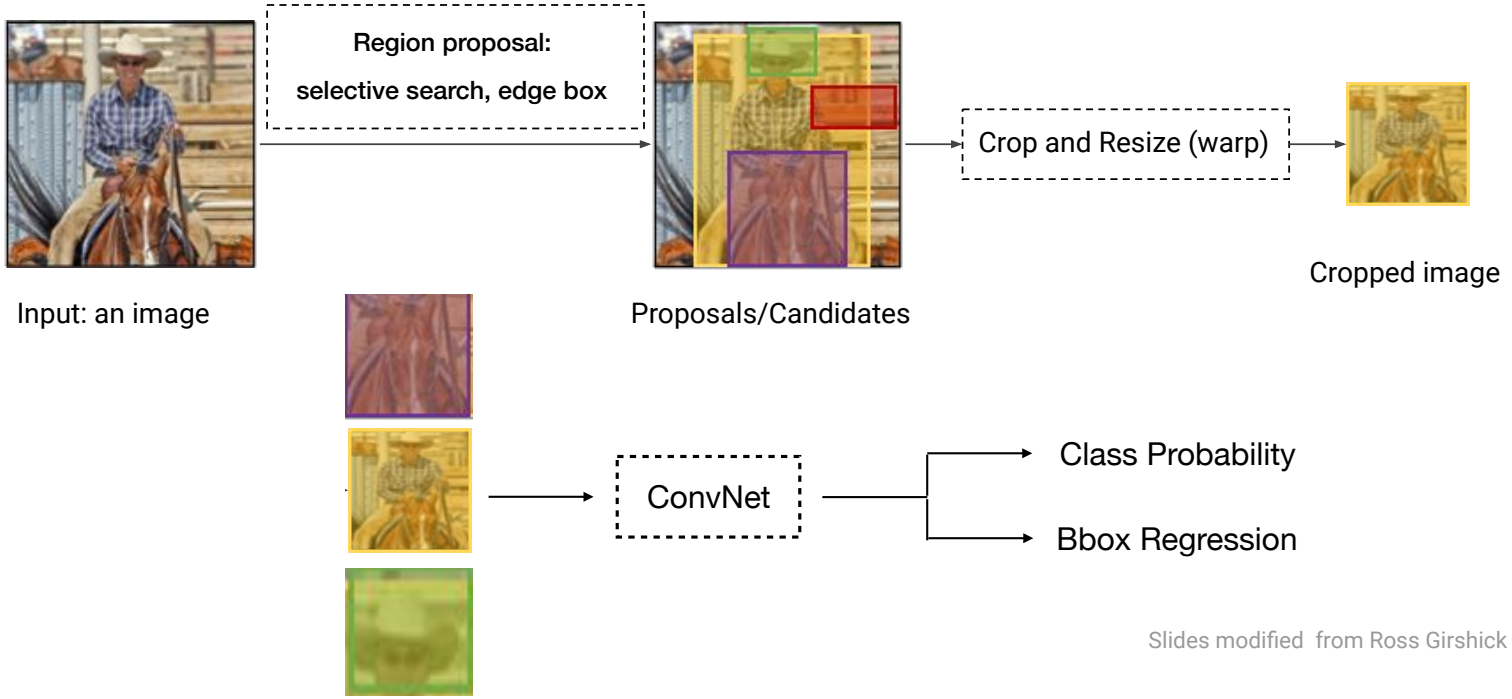
- Propose large number of regions potentially with objects
- Classify each proposed region



Slides modified from Ross Girshick tutorial at CVPR 2019

R-CNN Training

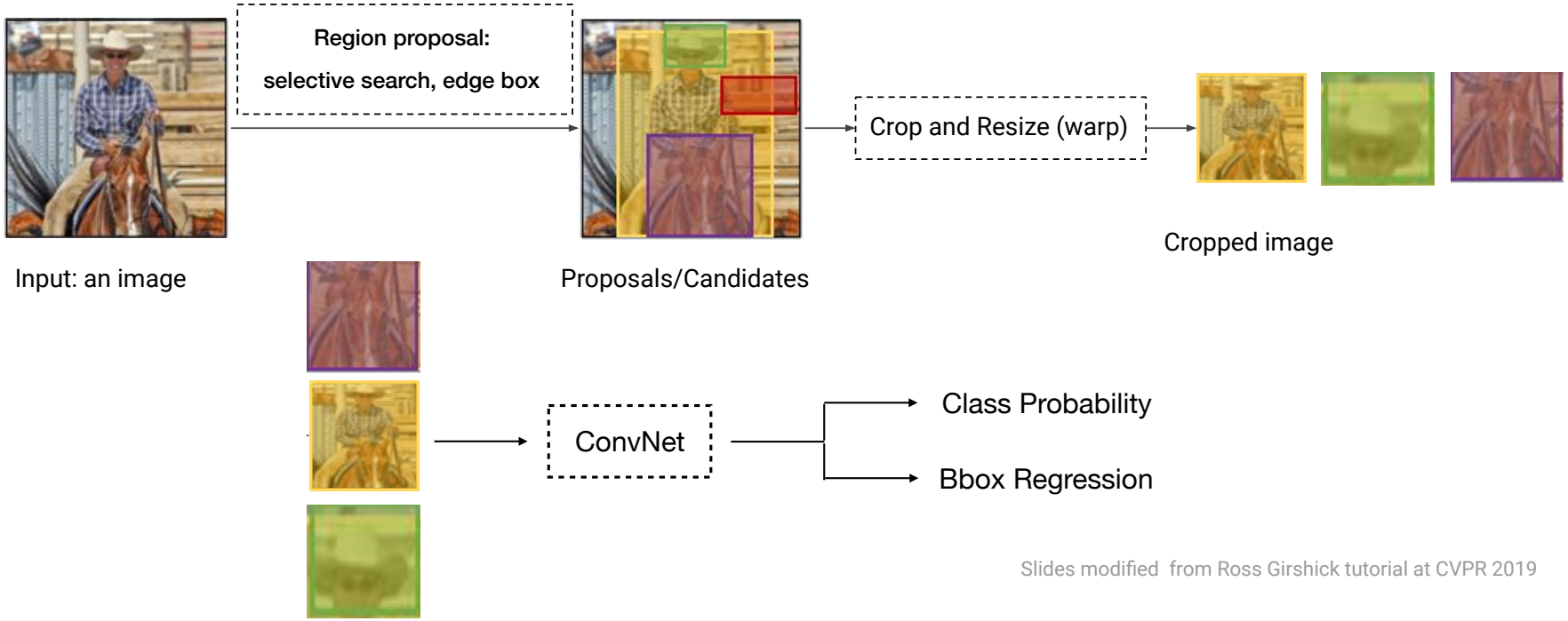
- Step 1: Train (or download) a classification model for ImageNet (AlexNet)



Slides modified from Ross Girshick tutorial at CVPR 2019

R-CNN Training

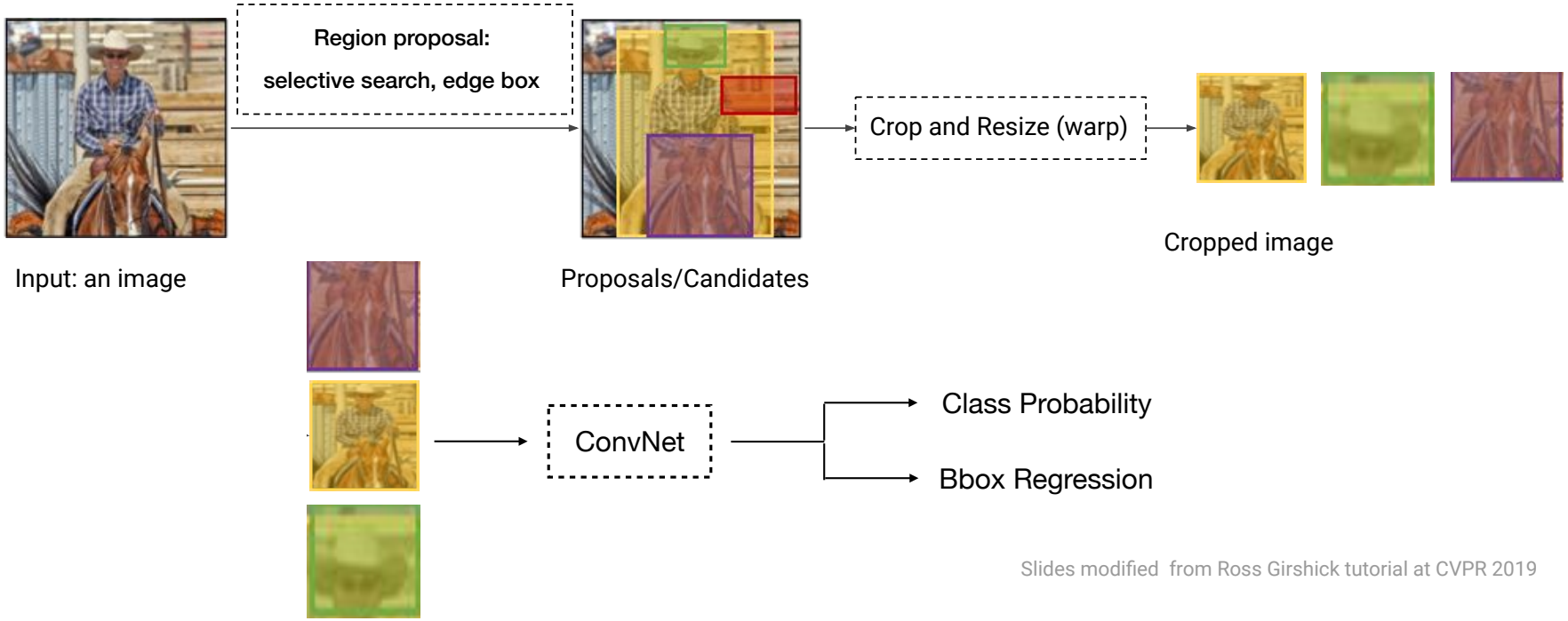
- Step 2: Fine-tune model for detection:
  - Instead of 1000 ImageNet classes → 20 object classes + 1 background
  - Throw away fc layer, re-initialize it
  - Input: Instead of images → Region Proposals (cropped and resized)



Slides modified from Ross Girshick tutorial at CVPR 2019

R-CNN Training

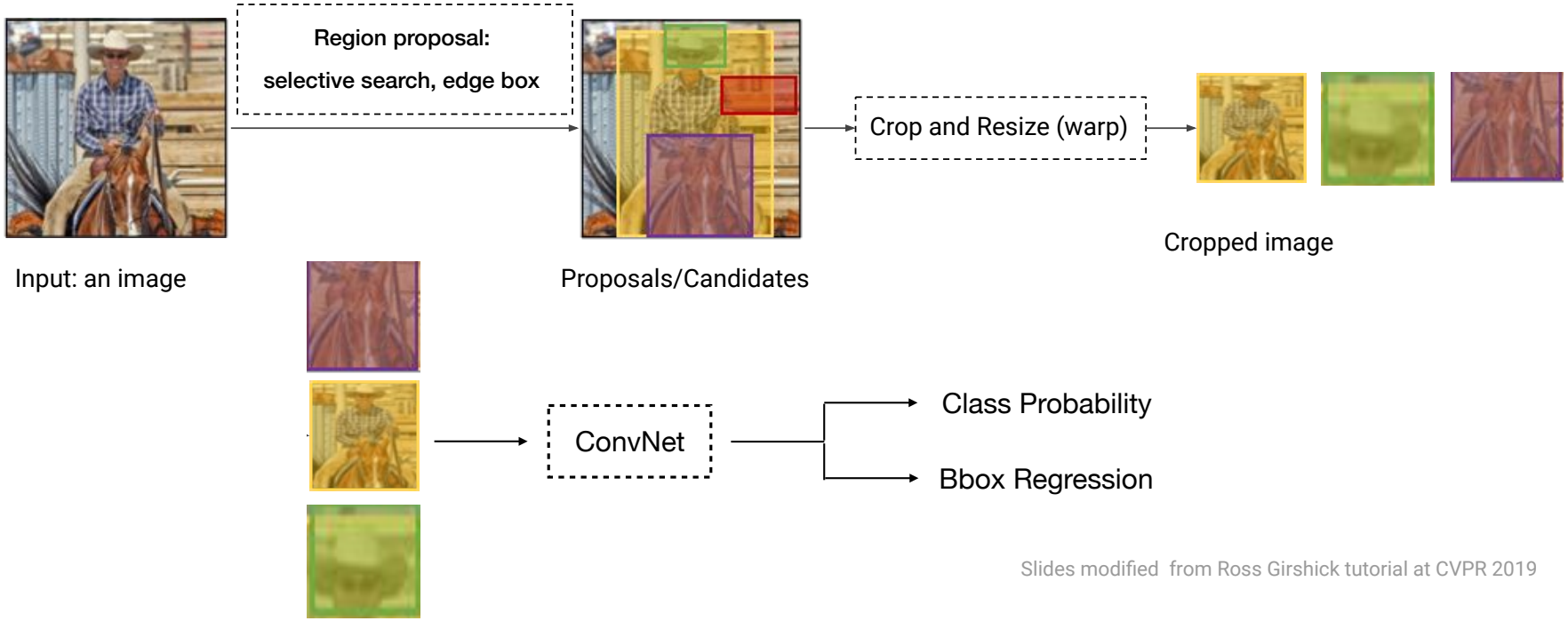
- Step 3: Extract features:
  - Input: Instead of images → Region Proposals (cropped and resized)
  - Save pool5 features to disk → ~100GB for a dataset of 10k images with 20 object classes (PASCAL VOC 2007)



Slides modified from Ross Girshick tutorial at CVPR 2019

R-CNN Training

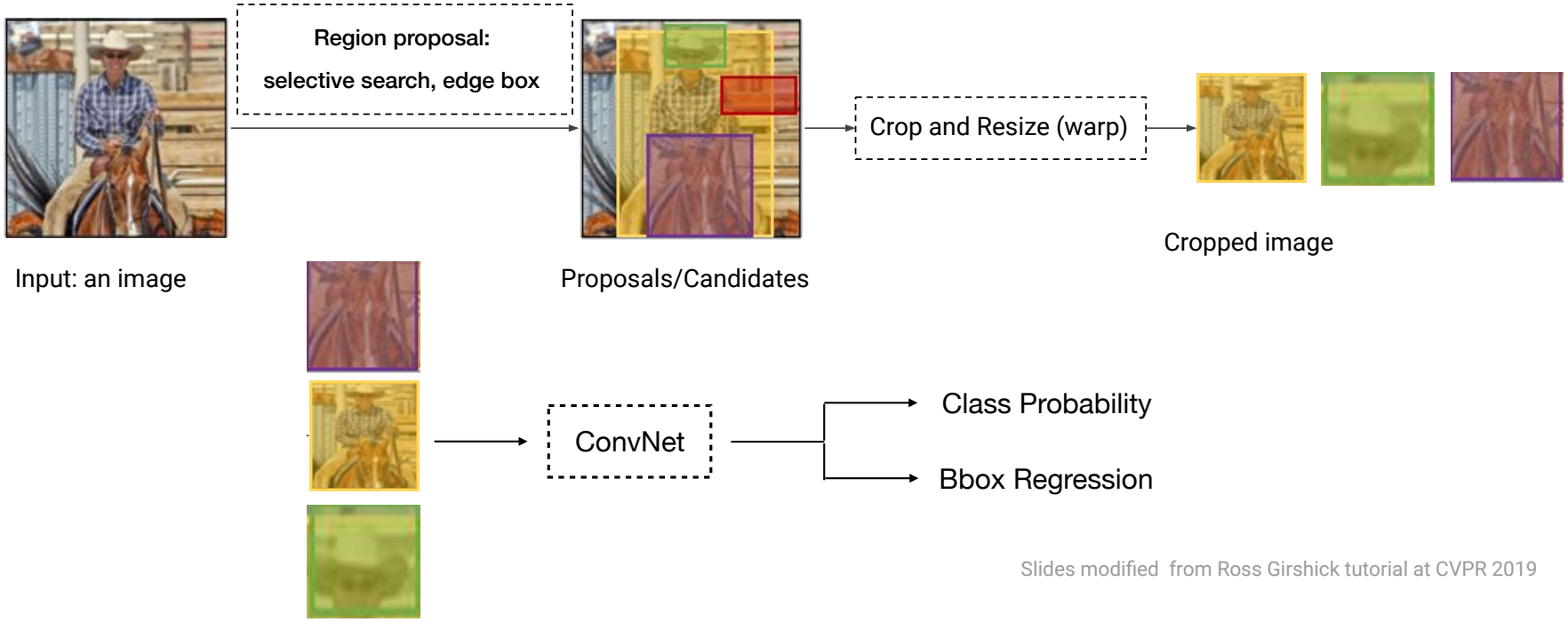
- Step 4: Train a binary SVM per class to classify region features



Slides modified from Ross Girshick tutorial at CVPR 2019

R-CNN Training

- Step 5: bounding-box regression:
  - For each class, train a linear regression model to map from features to offsets to ground-truth bounding boxes → makes up for ”slightly wrong” proposals



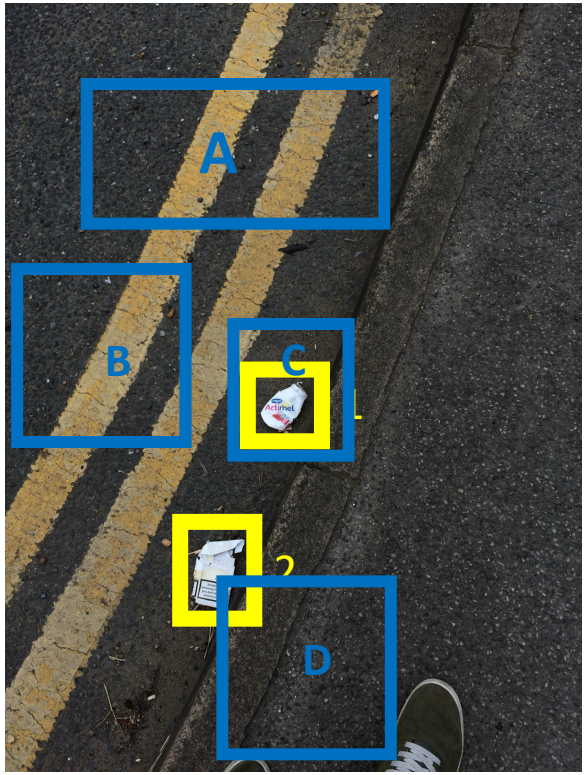
Slides modified from Ross Girshick tutorial at CVPR 2019

Basic Object Detector - Training

Basic Object Detector - Training

annotations

Basic Object Detector - Training

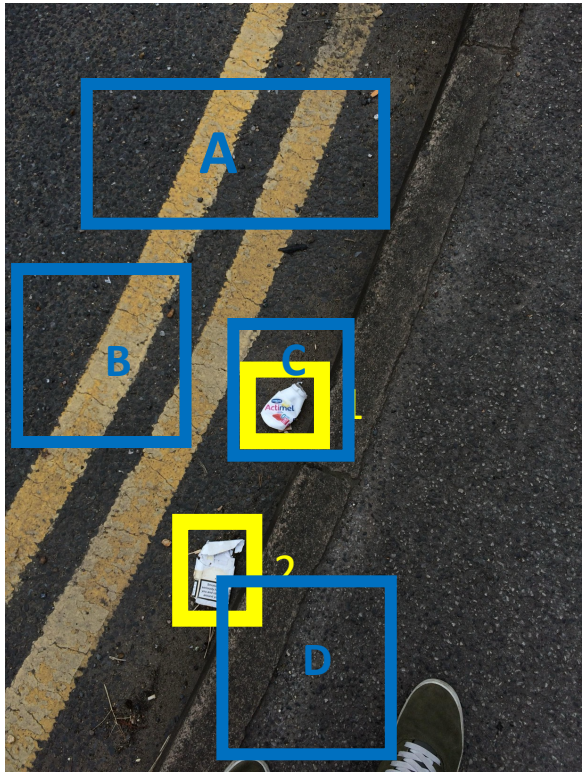


Run Selective Search, Edge Boxes, etc..

Tips:

- If slow, resize very large images beforehand (e.g. largest dimension 500)
- In case of SS, be sure that you use the ‘fast’ mode
- EdgeBoxes method is much faster

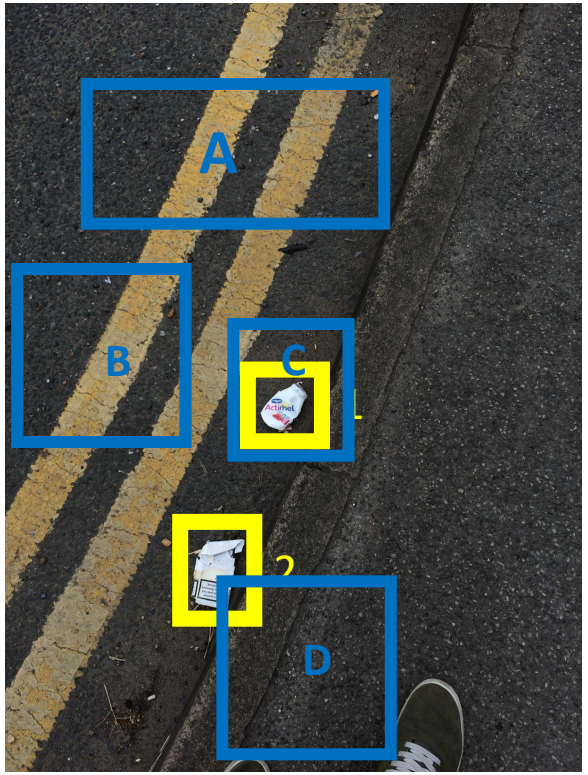
Basic Object Detector - Training



crop and resize (wrap)

Basic Object Detector - Training

x y



crop and resize (wrap)

A

B

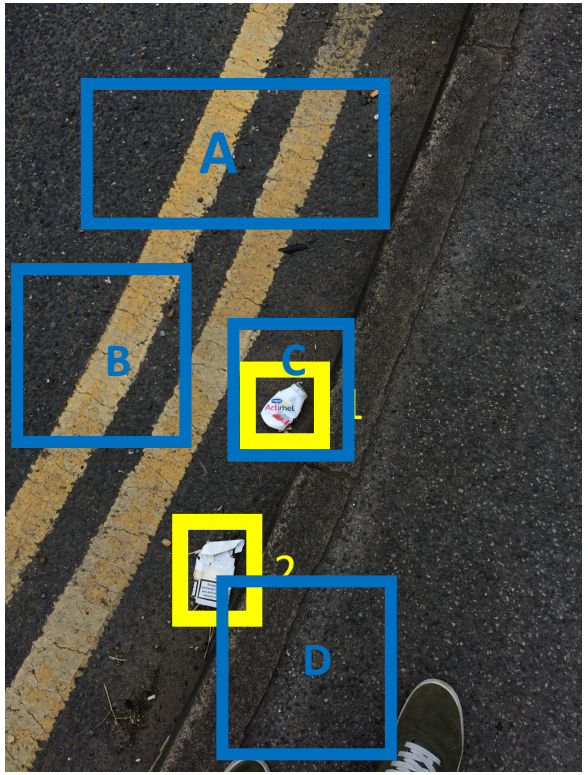
C

D

1

2

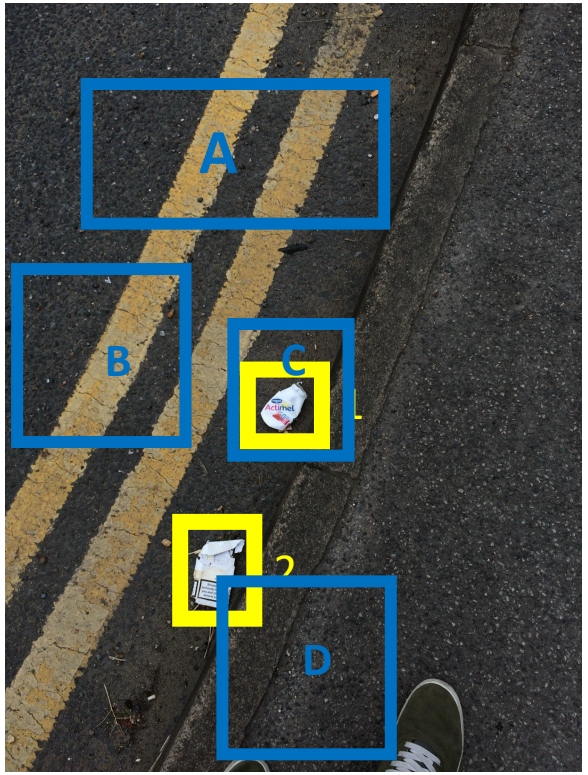
Basic Object Detector - Training



crop and resize (wrap)

	x	y
A		?
B		?
C		?
D		?
1		class1
2		class2

Basic Object Detector - Training



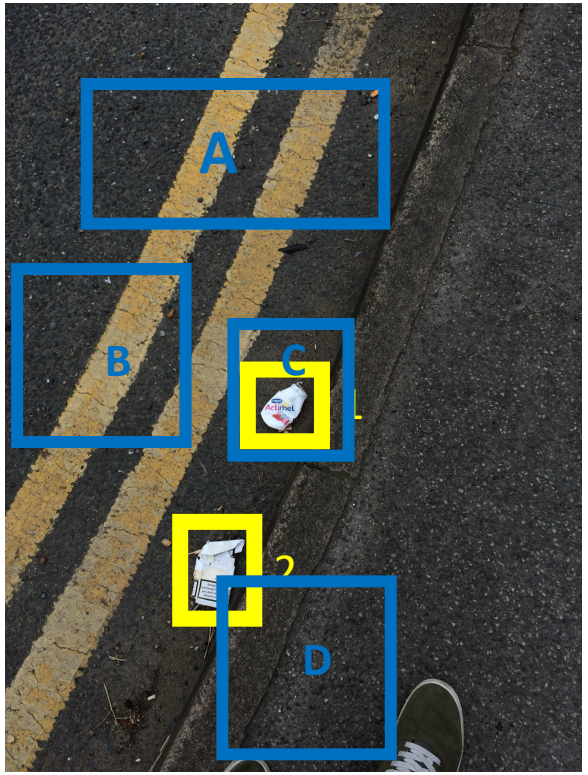
crop and resize (wrap)

	x	y
A		?
B		?
C		?
D		?
1		class1
2		class2

If $\max_i$ (IoU(A, GT_i)) < k1, then A is background

If $\max_i$ (IoU(A, GT_i)) >= k2, then A is class of GT_i

Basic Object Detector - Training



crop and resize (wrap)

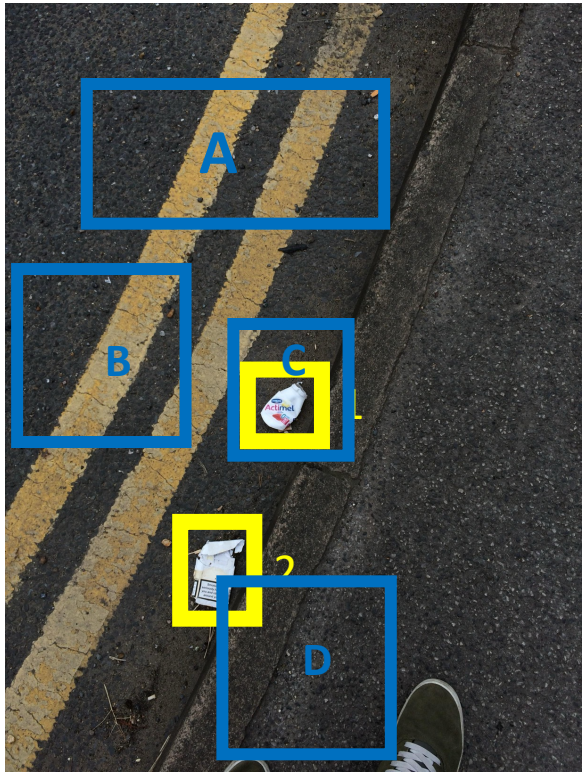
	x	y
A		background
B		background
C		class1
D		class2
1		class1
2		class2

If $\max_i$ (IoU(A, GT_i)) < k1, then A is background

If $\max_i$ (IoU(A, GT_i)) >= k2, then A is class of GT_i

- k1=k2=0.5
- k1=0.3, k2=0.5
- k1=0.3 k2=0.7

Basic Object Detector - Training



crop and resize (wrap)

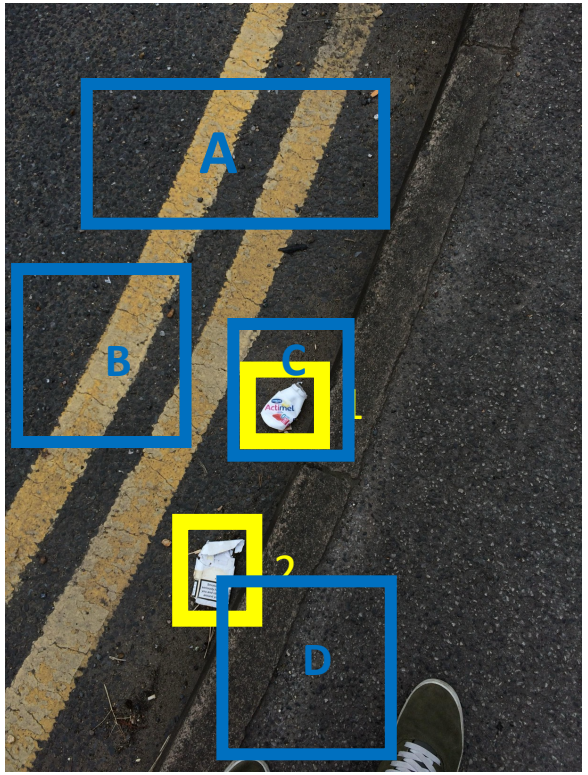
	x	y
A		background
B		background
C		class1
D		class2
1		class1
2		class2

CNN

Finetune a CNN pretrained on ImageNet to classify proposal windows

N+1 output classes

Basic Object Detector - Training



crop and resize (wrap)

	x	y
A		background
B		background
C		class1
D		class2
1		class1
2		class2

→ **CNN**

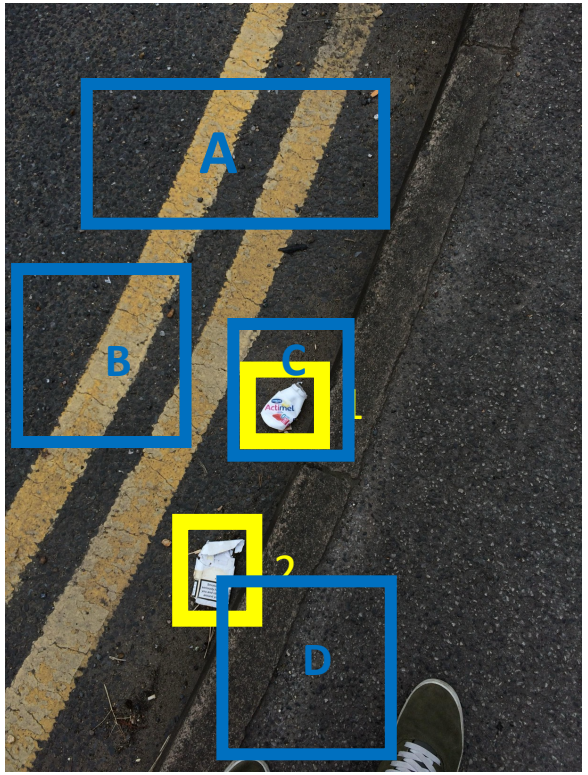
Finetune a CNN pretrained on ImageNet to classify proposal windows

N+1 output classes

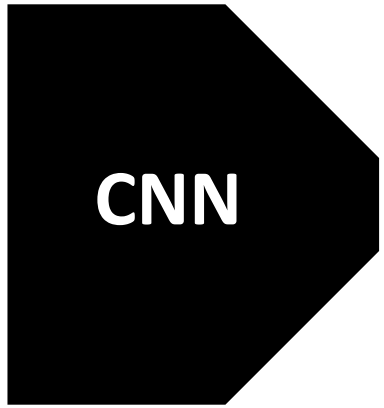
- Be careful with the class imbalance: >>99% of proposals will be "background"

- Sample proposals so that you train with 75% background and 25% of any class

Basic Object Detector - Training



crop and resize (wrap)



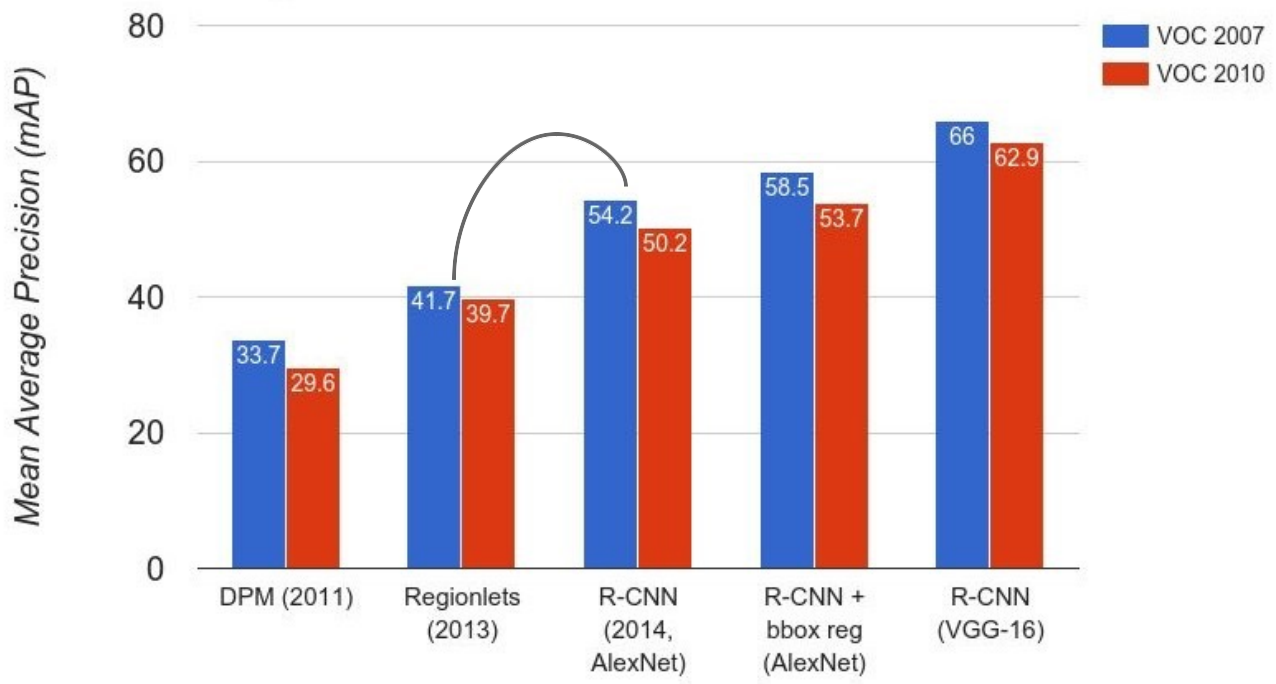
Finetune a CNN pretrained on ImageNet to classify proposal windows

N+1 output classes

- Be careful with the class imbalance: >>99% of proposals will be "background"

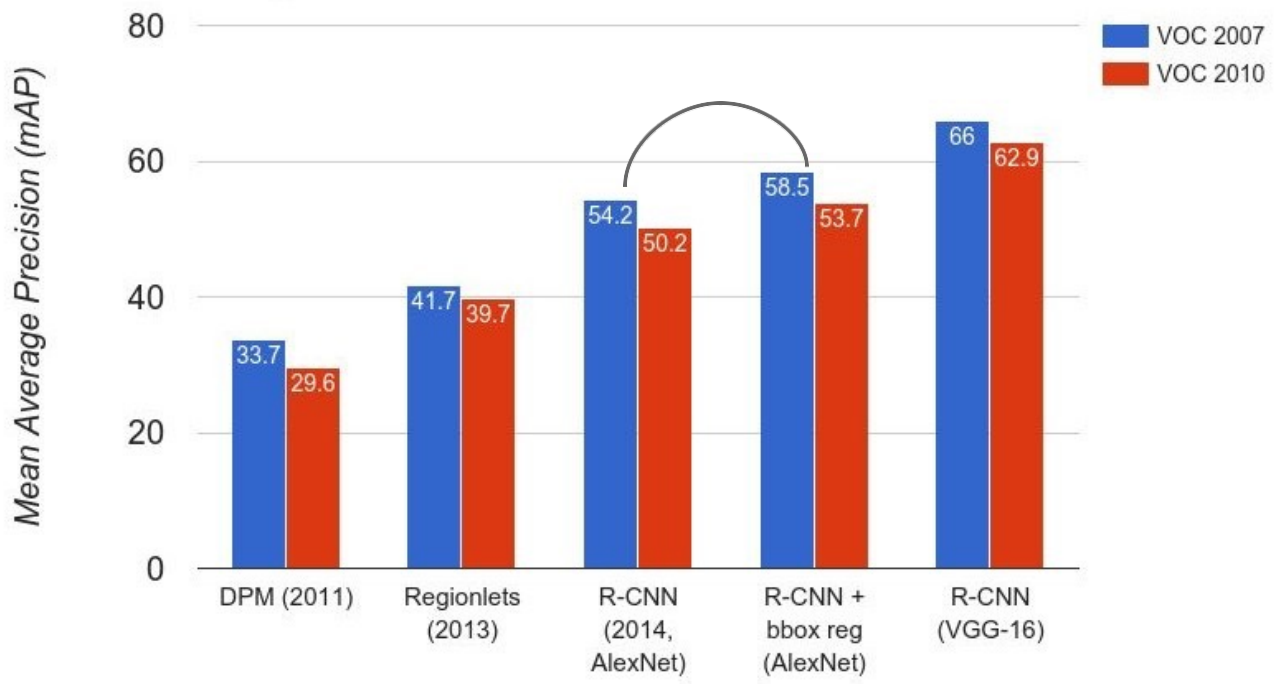
- Sample proposals so that you train with 75% background and 25% of any class (e.g. batch:64, 38 background windows, 16 "positive"

R-CNN Results



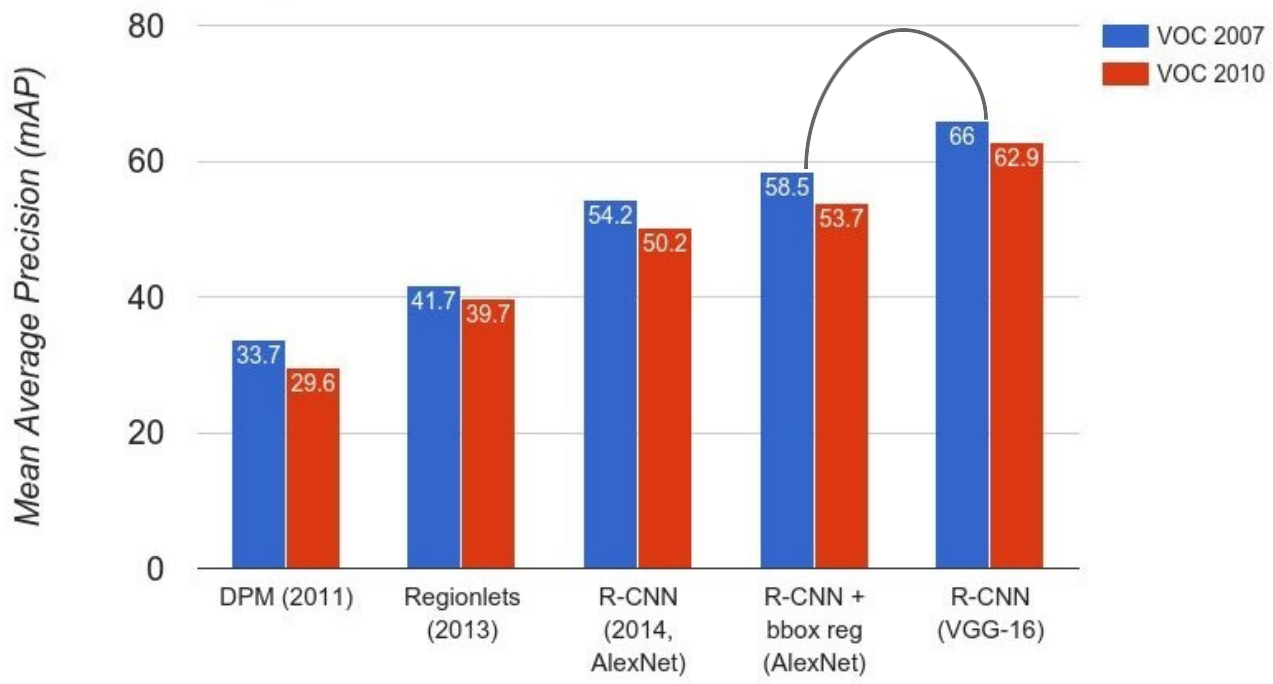
Big improvement compared to pre-CNN methods

R-CNN Results



Big improvement compared to pre-CNN methods
Bounding-box regression helps

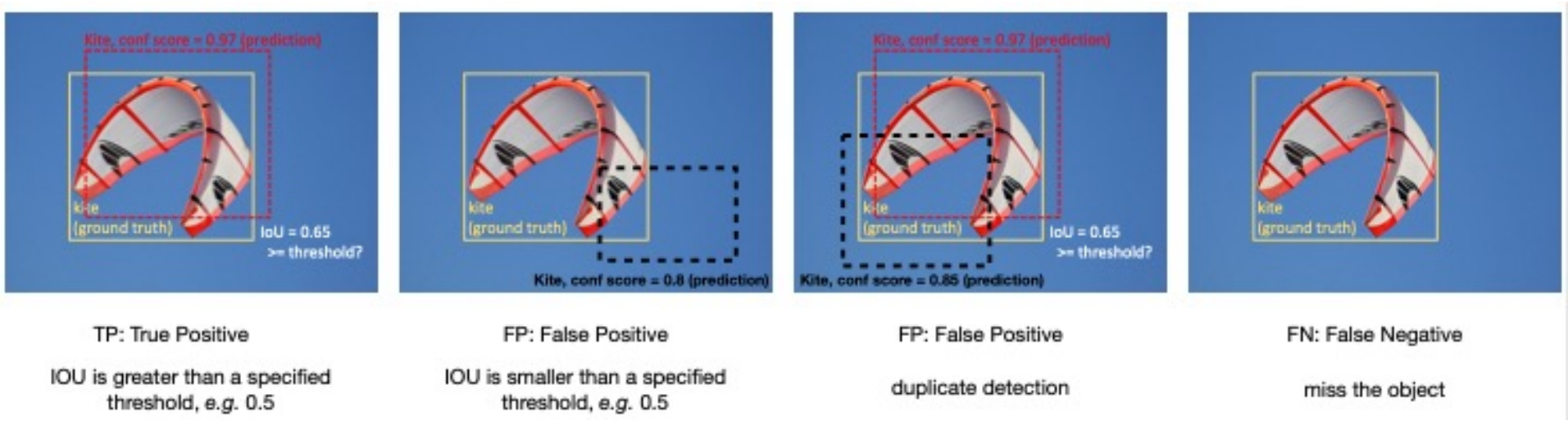
R-CNN Results



Big improvement compared to pre-CNN methods
Bounding-box regression helps
Features from deeper network help

Object Detection Terminology

- The model's prediction is bounding boxes with category confidence scores, *e.g.* person, dog, background, *etc*
- **IoU**: intersection over union between a pair of boxes, *aka.* **Jaccard Similarity**
- **Goal**: large IOU, high confidence score for the correct category



Questions??

Exercise 3.1

Detecting potholes in the wild

Build an object detector from scratch step-by-step!

Exercise 3.1

World first AI robot for tackling the pothole problem

Subscribe to UKRI emails

Sign up for news, views, events and funding alerts.

Email address

Subscribe

21 November 2023

At Daresbury Laboratory, UK start-up Robotiz3d is combining artificial intelligence (AI) with advanced robotics to tackle the pothole problem.

Exercise 3.1

Detecting potholes in the wild

Tasks for simple object detector:

- Extract object proposals
- Evaluate the object proposals
- Determine the require number of proposals per image
- Prepare the proposals for the training of the object detector

Improve road safety: Detecting potholes in the wild

Project 3.1 - Object proposals

Introduction to Deep Learning in Computer Vision

January 2024

A pothole is a depression in a road surface, usually asphalt pavement, where traffic has removed broken pieces of the pavement. It is usually the result of water in the underlying soil structure and traffic passing over the affected area. Potholes have a great impact on the road safety.

In this project, you are asked to build a deep-learning object detection system that can automatically detect potholes in images in the wild. This object detection can then be deployed in robotic machines or cars that can scan areas and improve the condition of roads. Detecting potholes in the wild can be a very challenging problem.

Figure 1: Examples from the Potholes dataset.

Exercise 3.1

Annotations XML Pascal format

```
<annotation>
    <folder>dataset</folder>
    <filename>img-655.jpg</filename>
    <path>/CSE-800/Thesis/2020-01-23/dataset/img-655.jpg</path>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>534</width>
        <height>300</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>pothole</name>
        <pose>Unspecified</pose>
        <truncated>1</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>1</xmin>
            <ymin>187</ymin>
            <xmax>188</xmax>
            <ymax>276</ymax>
        </bndbox>
    </object>
    <object>
        <name>pothole</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>102</xmin>
            <ymin>84</ymin>
            <xmax>283</xmax>
            <ymax>151</ymax>
        </bndbox>
    </object>
    <object>
        <name>pothole</name>
        <pose>Unspecified</pose>
        <truncated>1</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>1</xmin>
            <ymin>55</ymin>
            <xmax>144</xmax>
            <ymax>133</ymax>
```

Feedback ☺

Join at menti.com | use code 6105 6425

Thank you!!!