

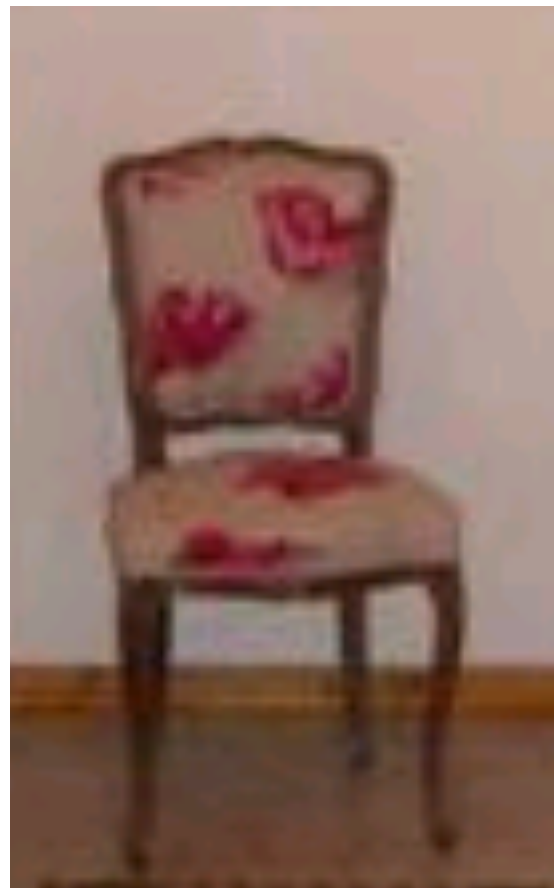
Lecture 13

Scene understanding

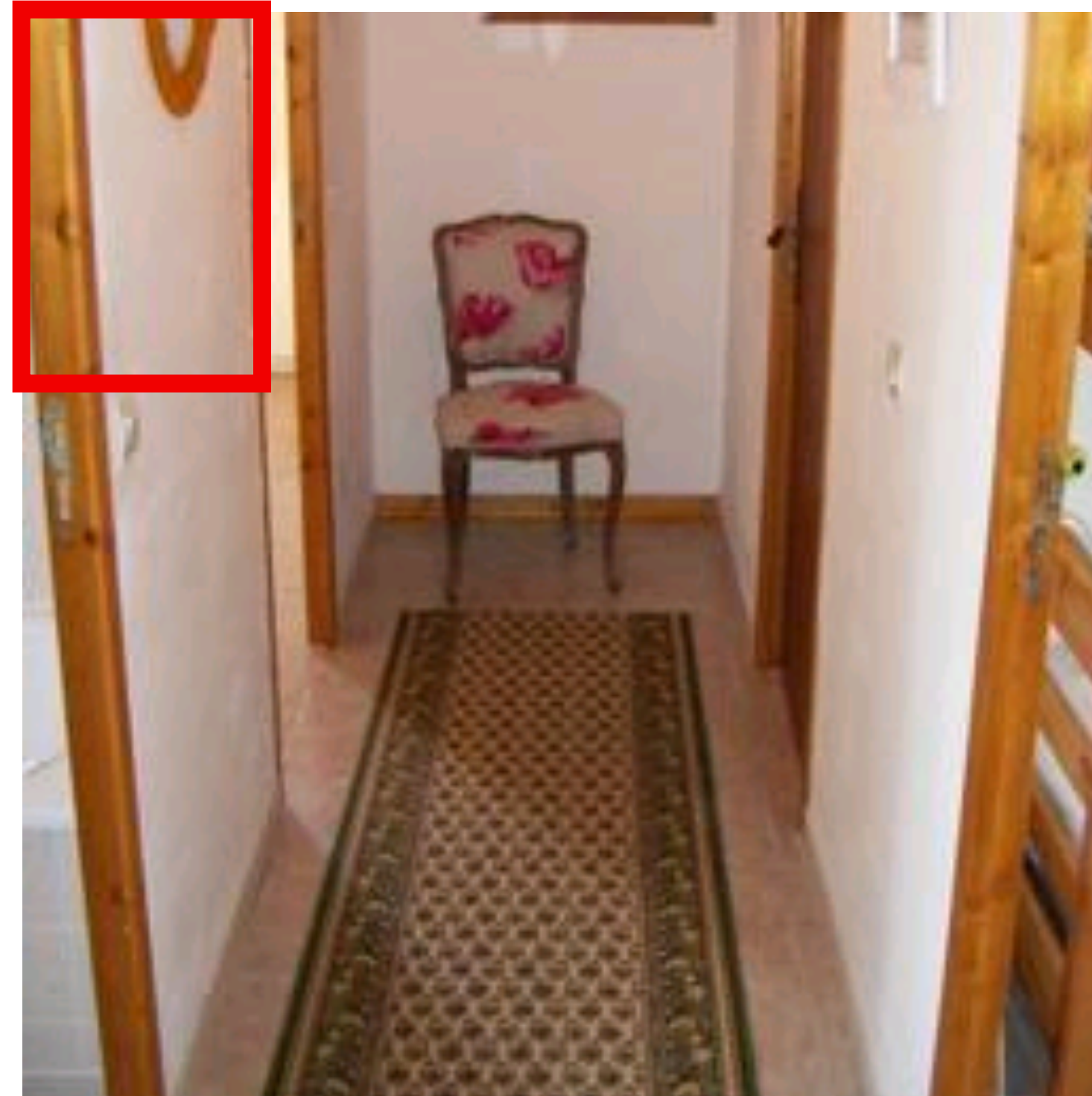
Object recognition

Is it really so hard?

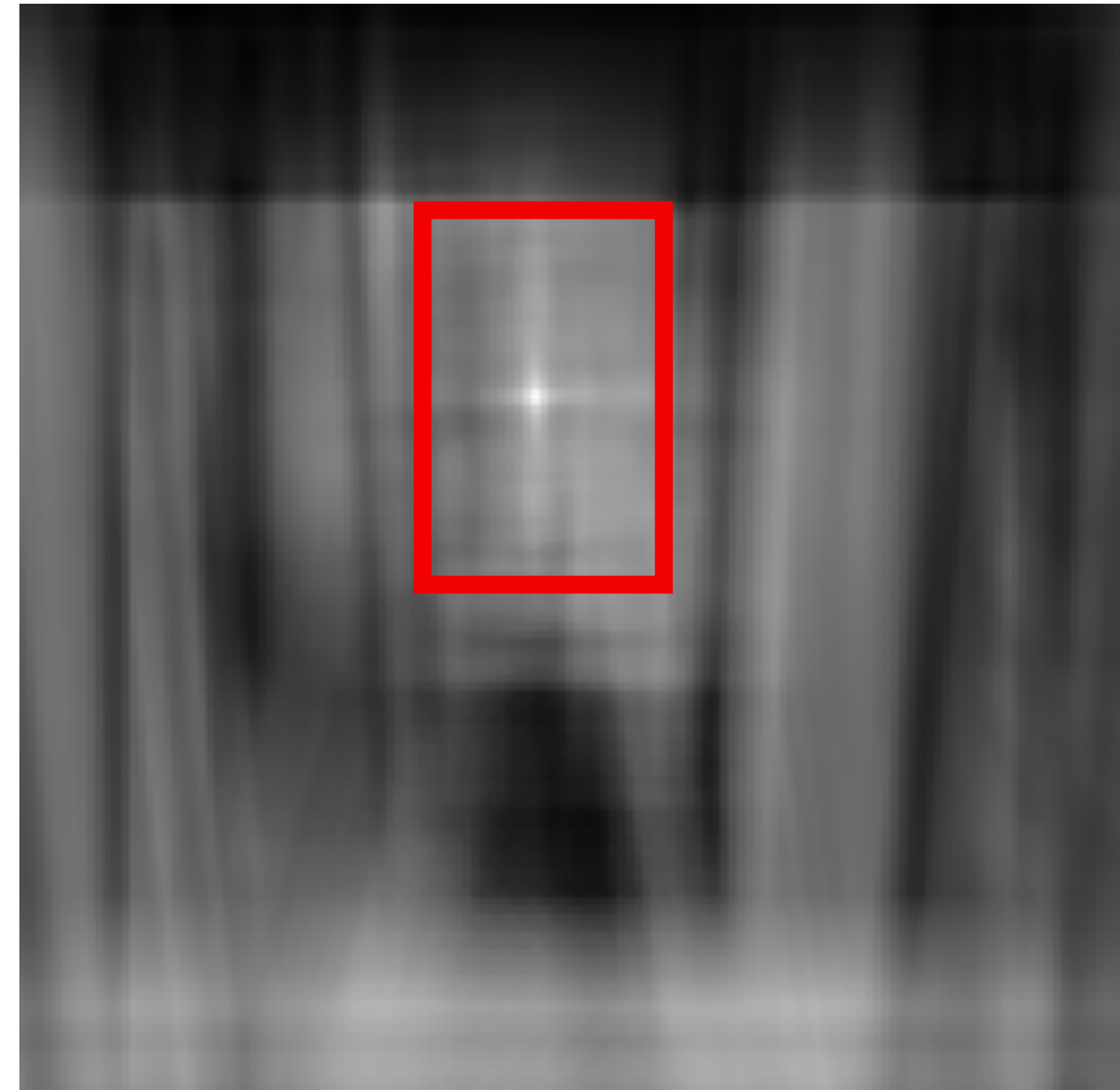
This is a chair

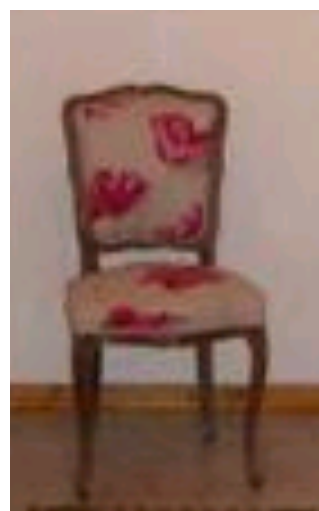


Find the chair in this image



Output of normalized correlation

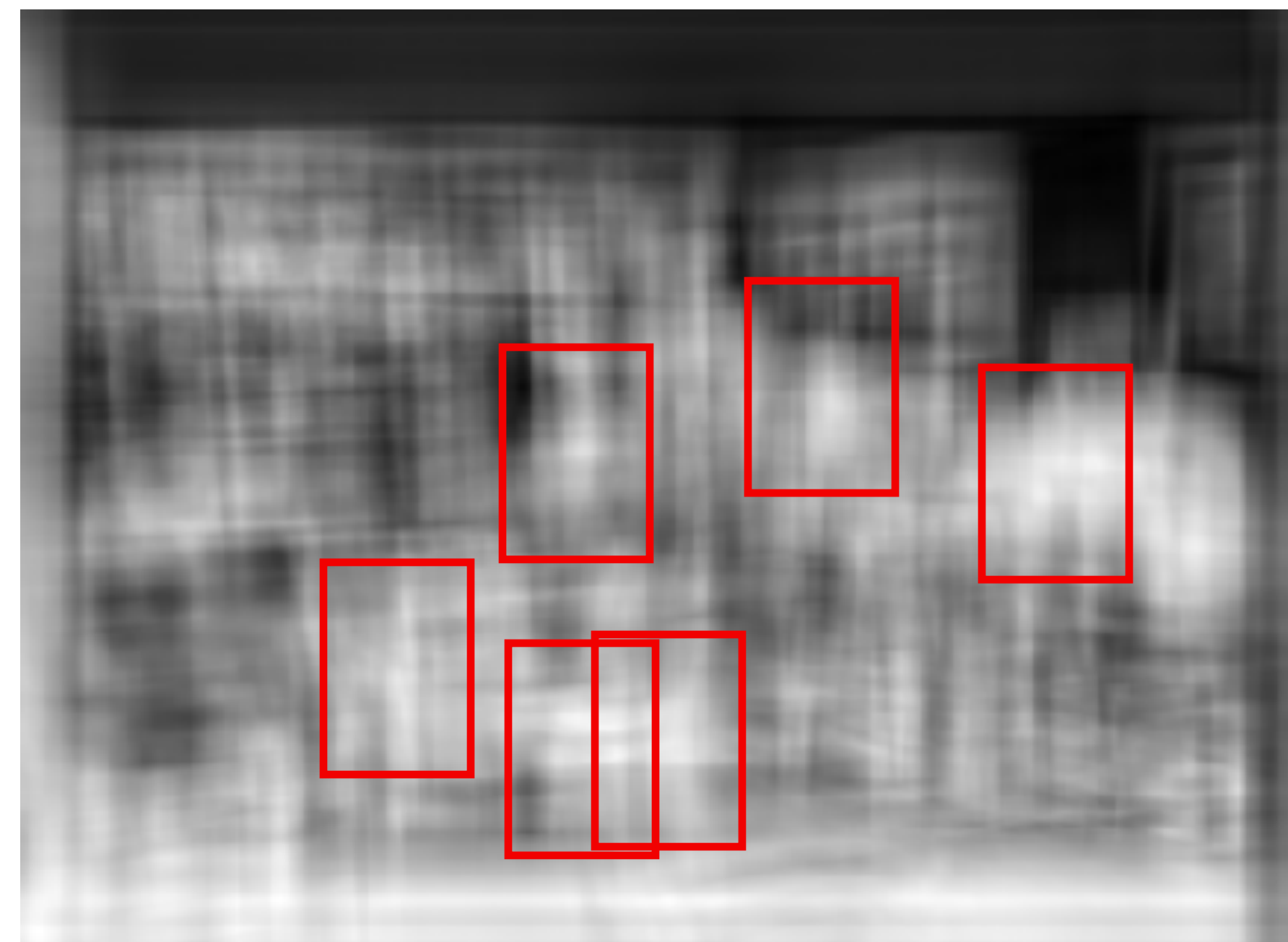
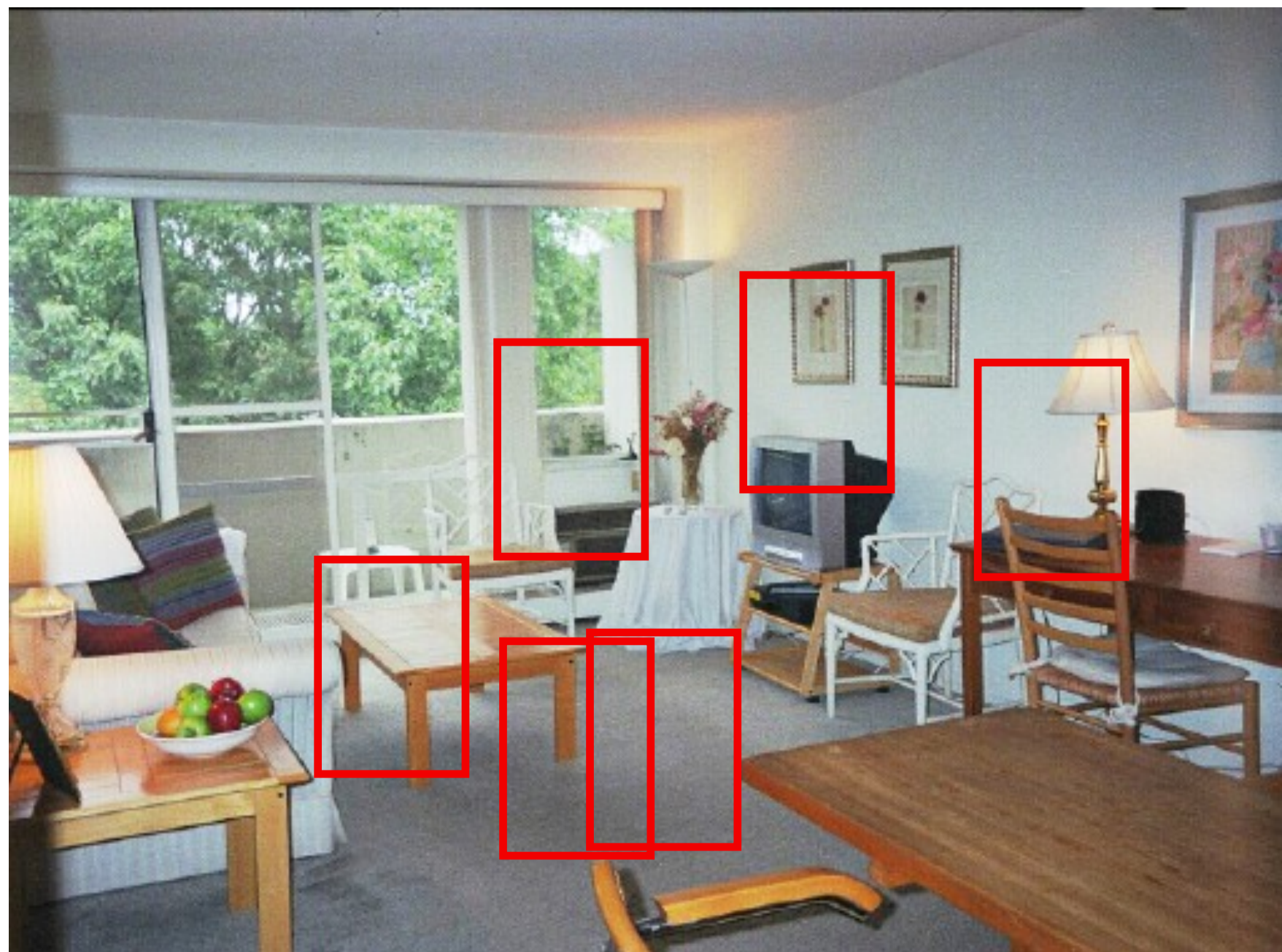




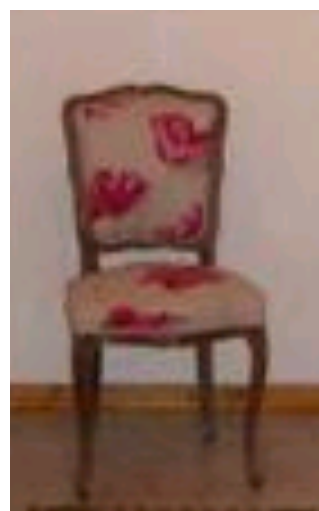
Object recognition

Is it really so hard?

Find the chair in this image



Pretty much random detections
Simple template matching is not going to make it



Object recognition

Is it really so hard?

Find the chair in this image



A “popular method is that of template matching, by point to point correlation of a model pattern with the image pattern. These techniques are inadequate for three-dimensional scene analysis for many reasons, such as occlusion, changes in viewing angle, and articulation of parts.” Nivatia & Binford, 1977.

Instances vs. categories

Instance matching

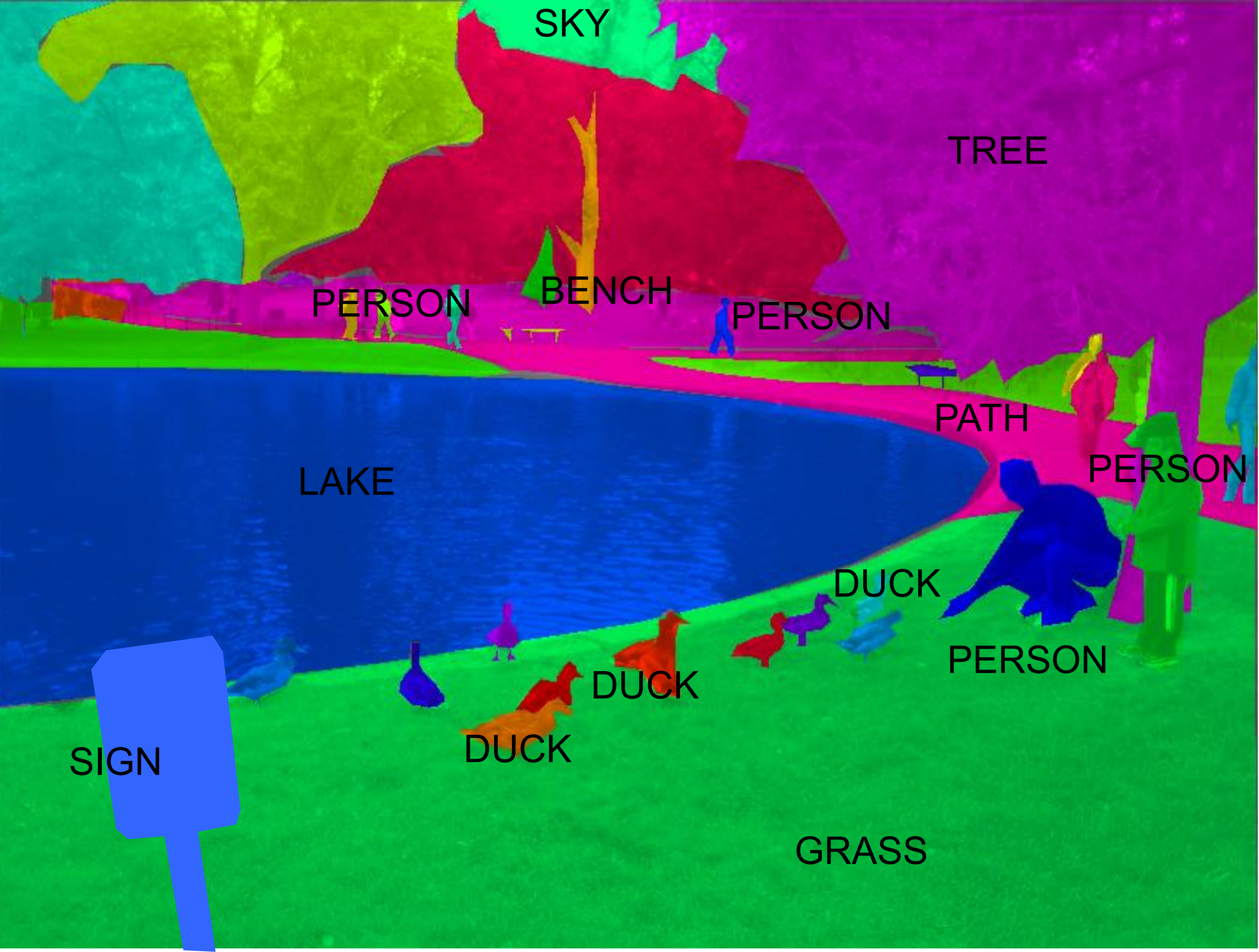
Find these two toys



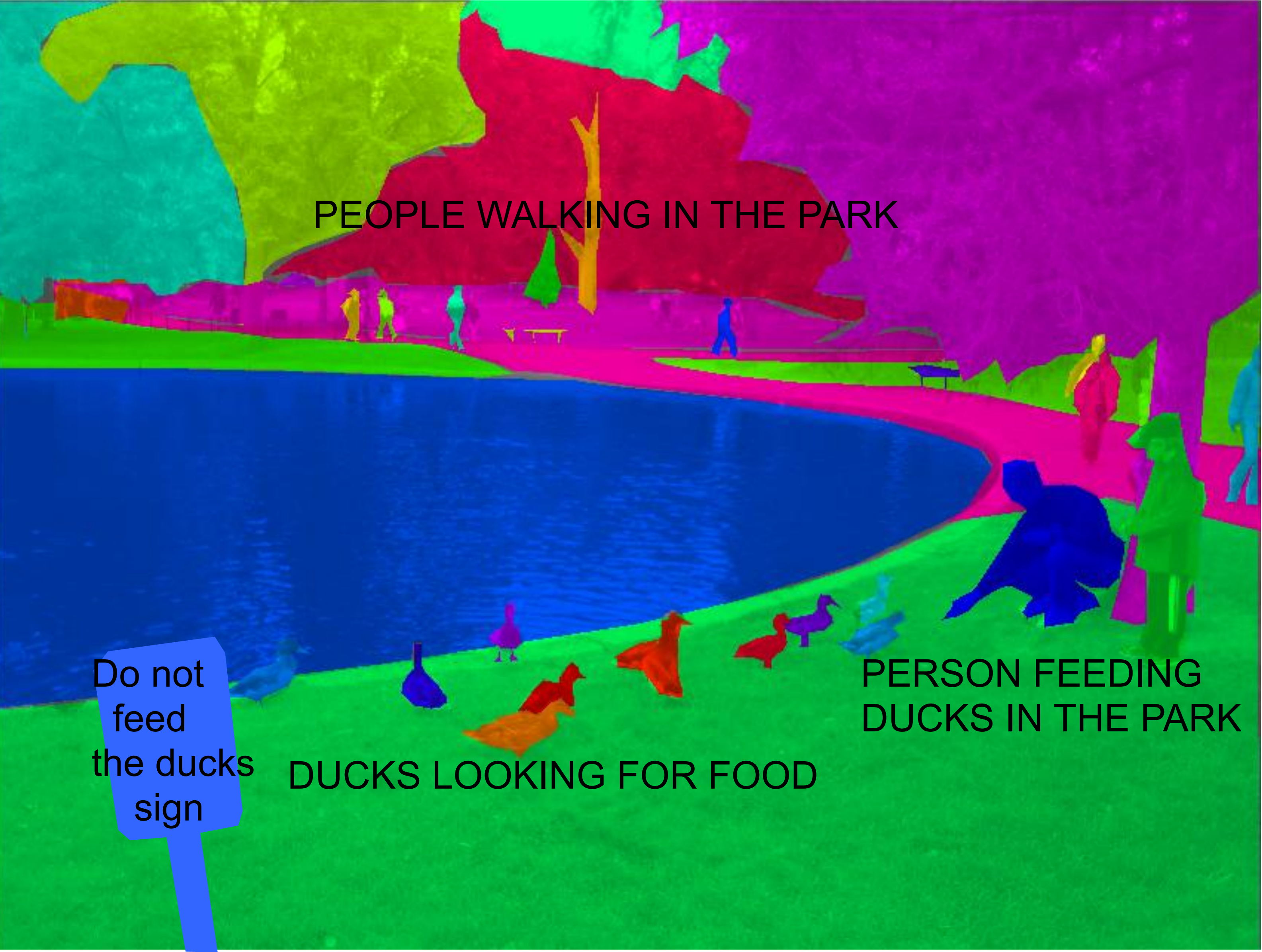
Categories

Find a bottle:









PEOPLE WALKING IN THE PARK

PERSON FEEDING
DUCKS IN THE PARK

DUCKS LOOKING FOR FOOD

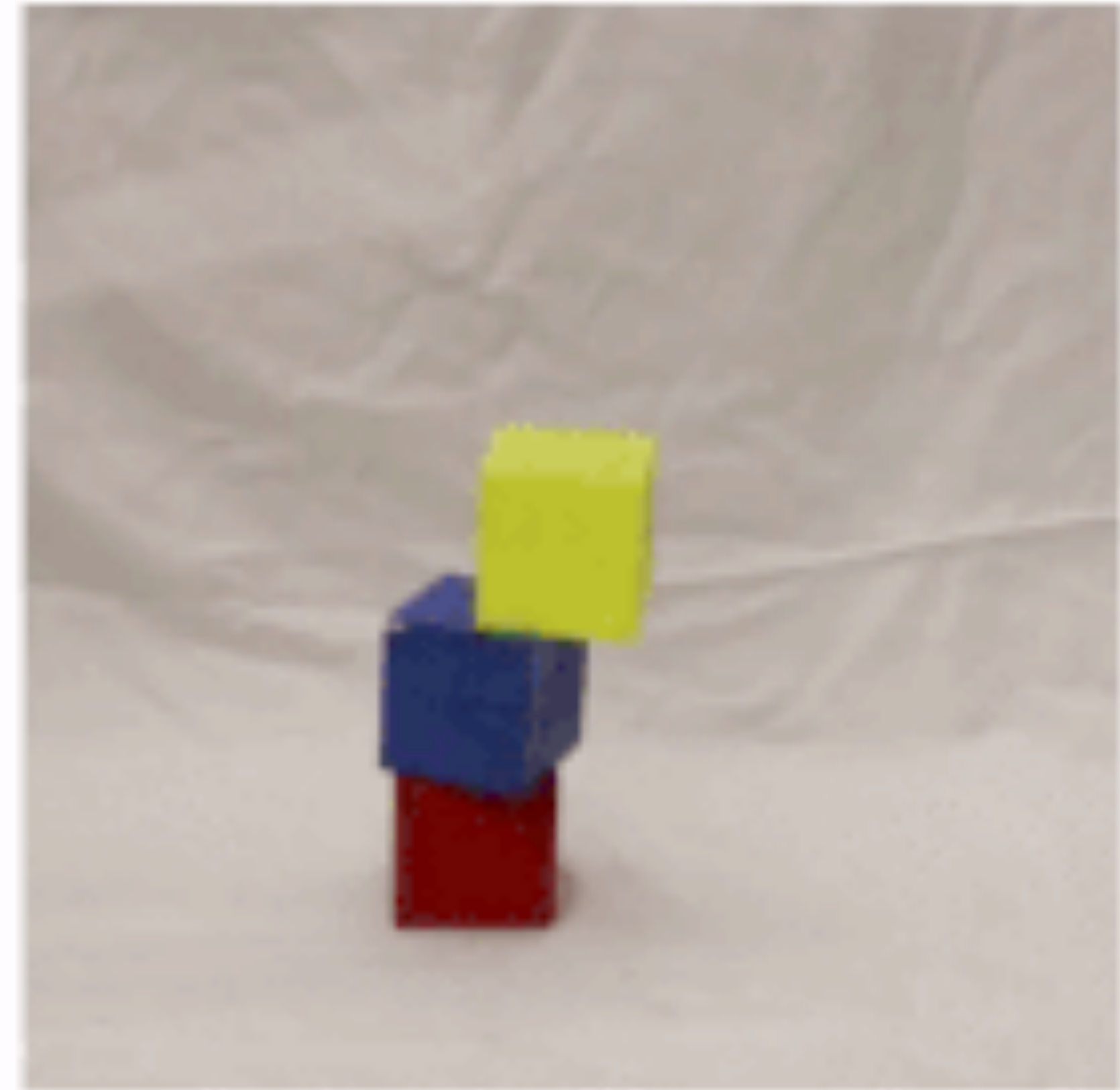
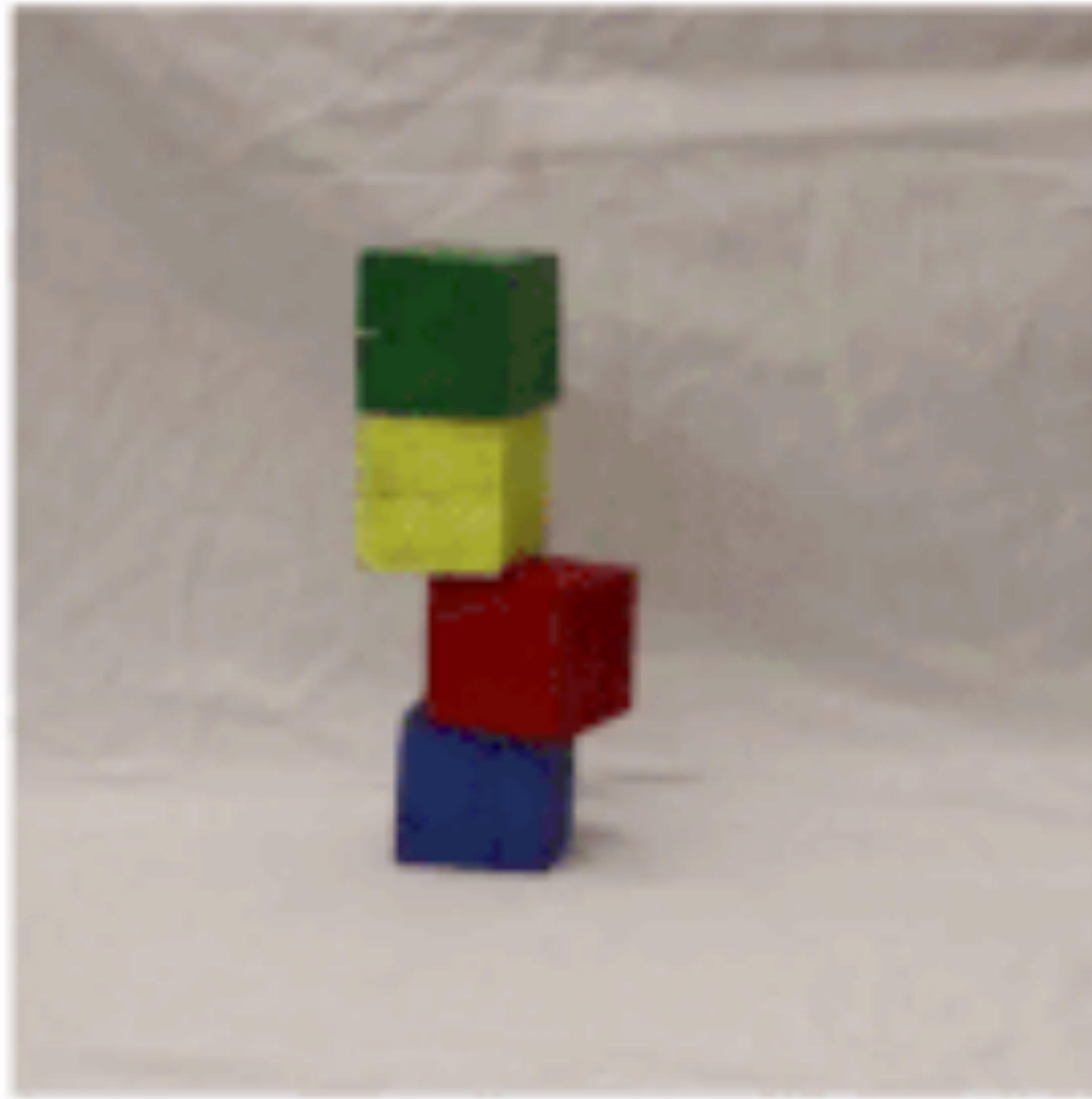
Do not
feed
the ducks
sign



PEOPLE UNDER THE
SHADOW OF THE TREES

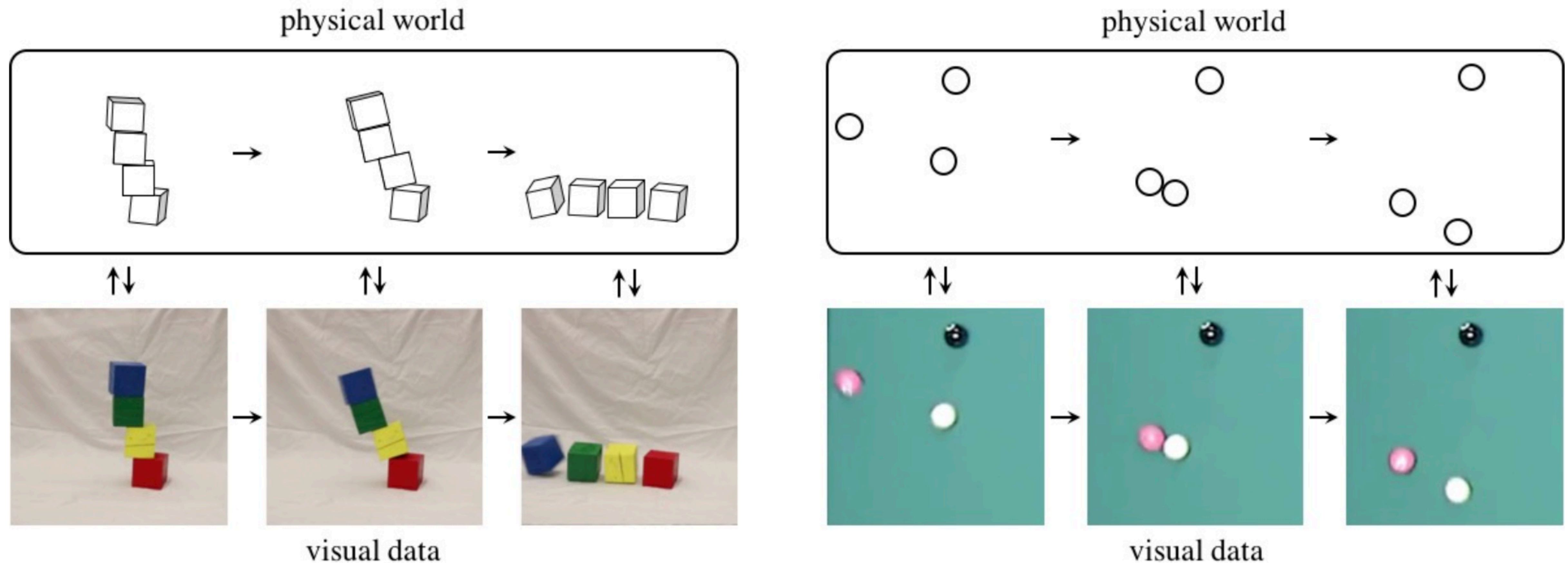
DUCKS ON TOP
OF THE GRASS

Intuitive physics



["Learning to See Physics via Visual De-animation", Wu et al., NIPS 2017]

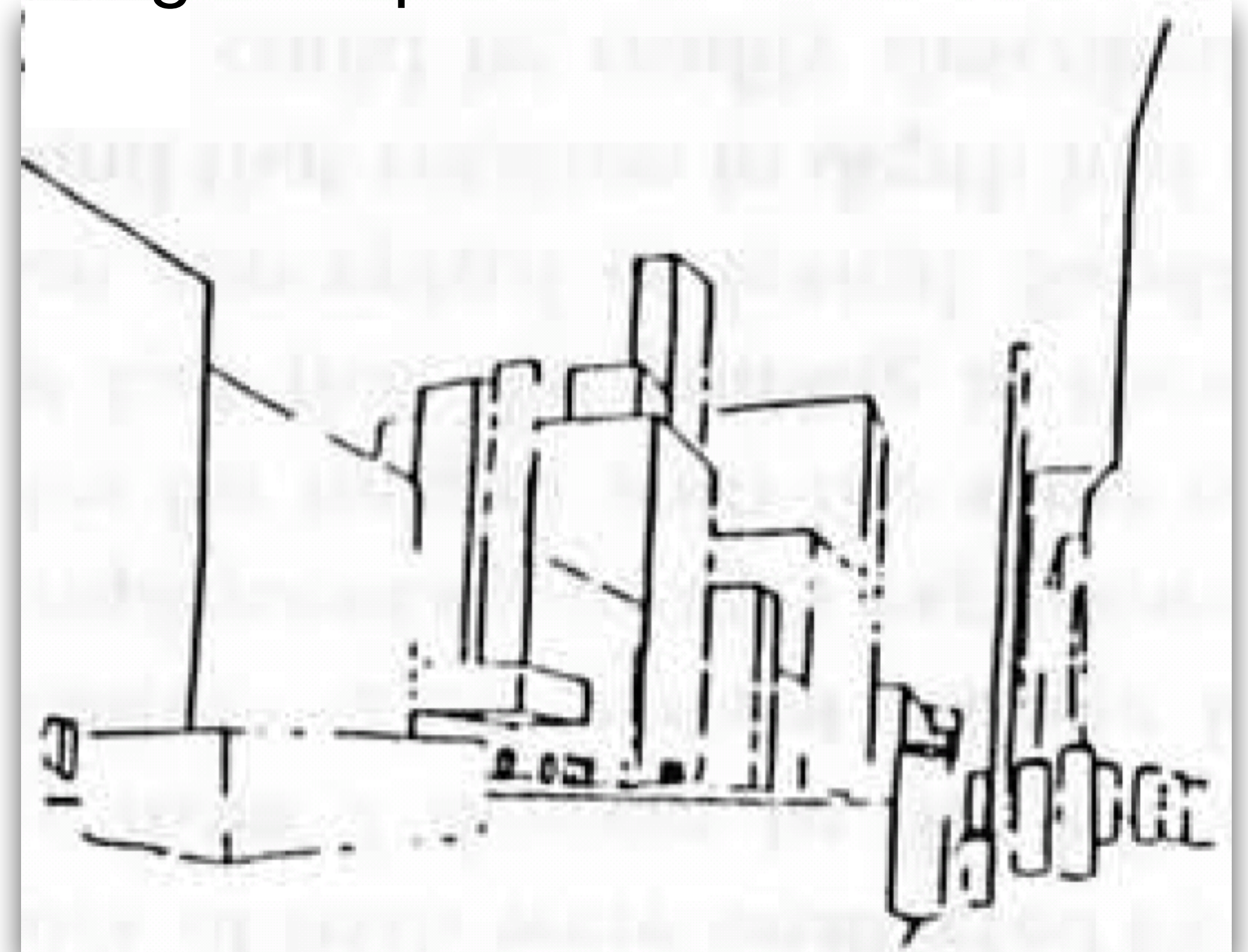
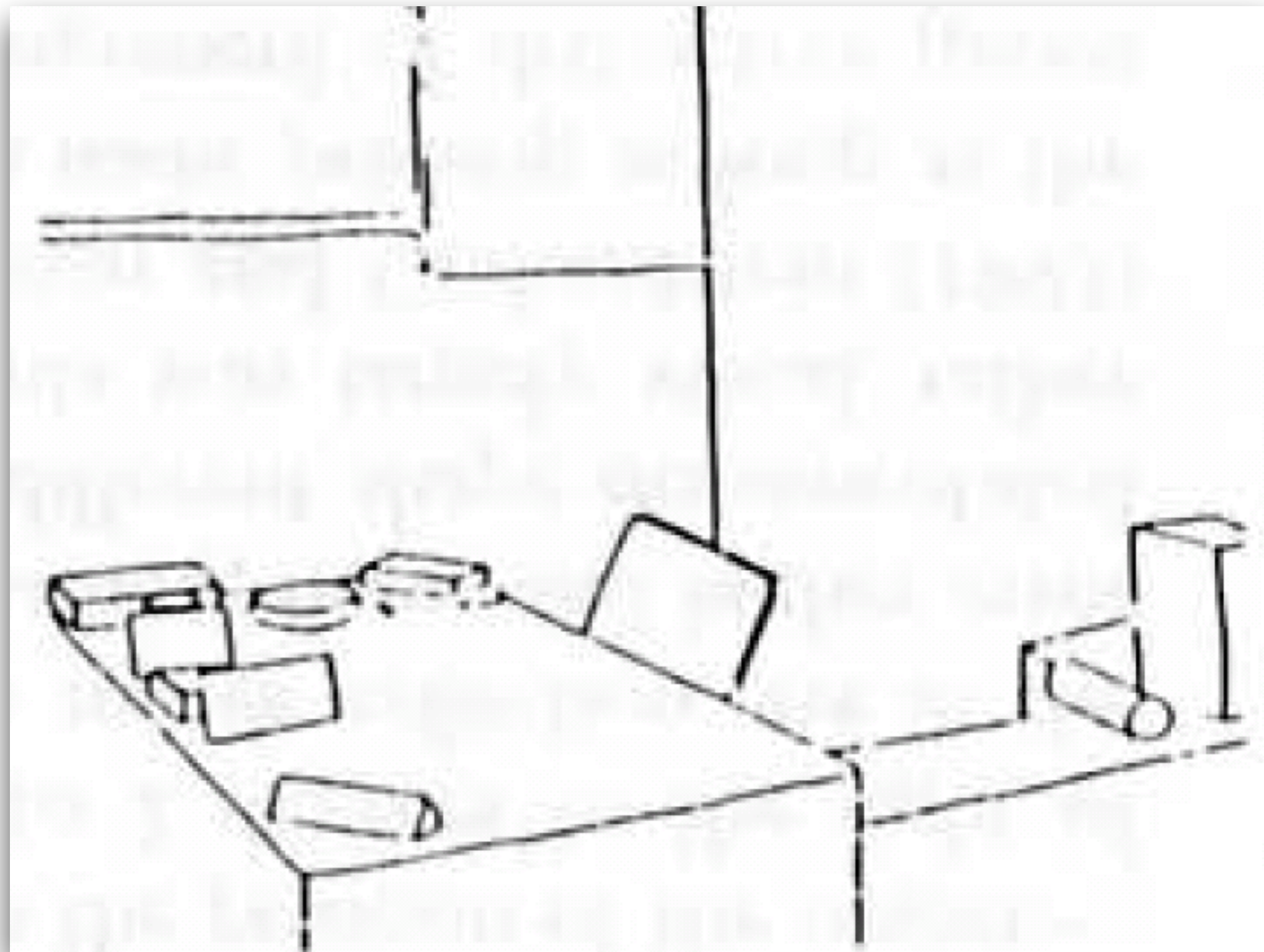
Intuitive physics



["Learning to See Physics via Visual De-animation", Wu et al., NIPS 2017]

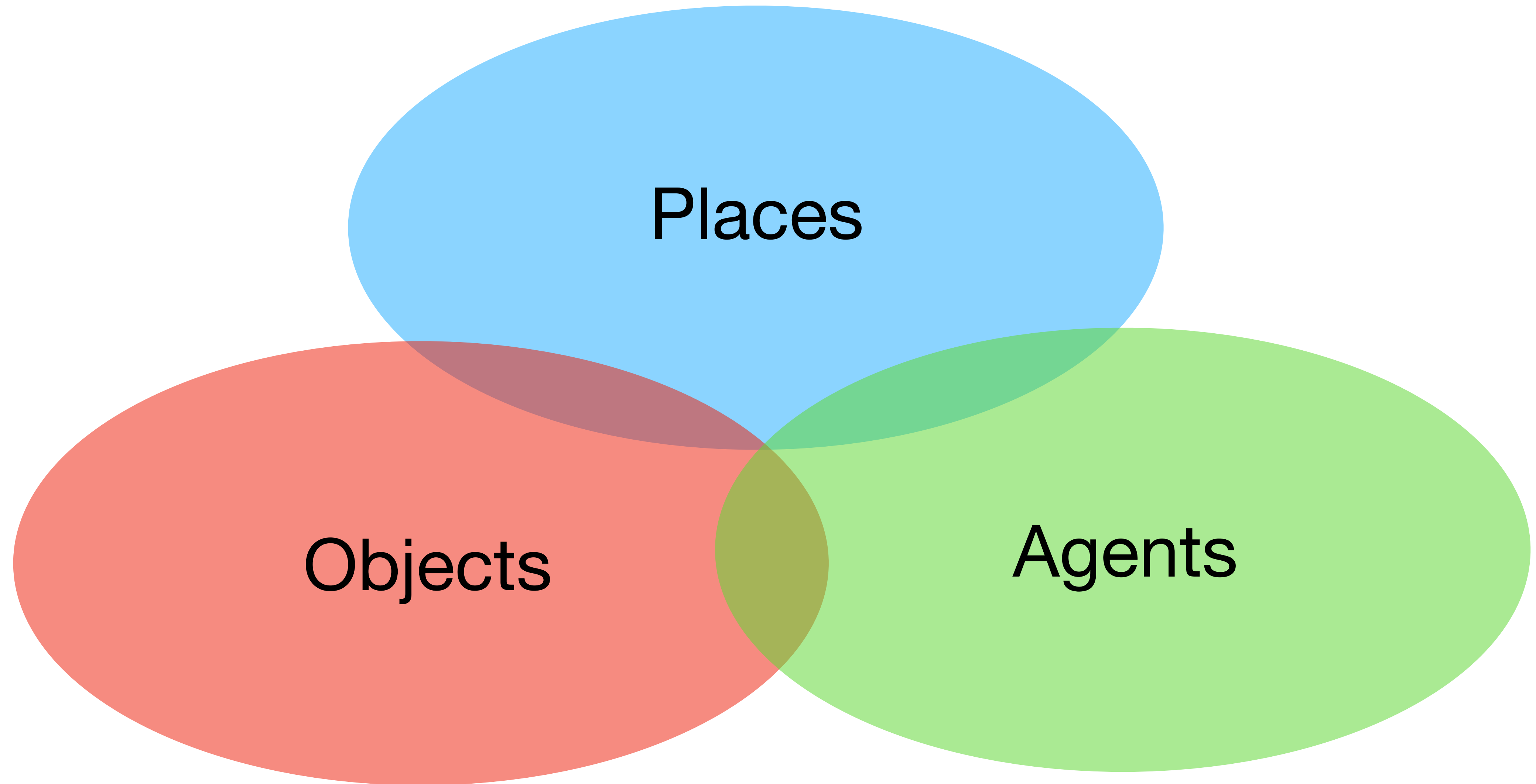
Scene understanding

Scene understanding is an integrated process



Mezzanotte & Biederman

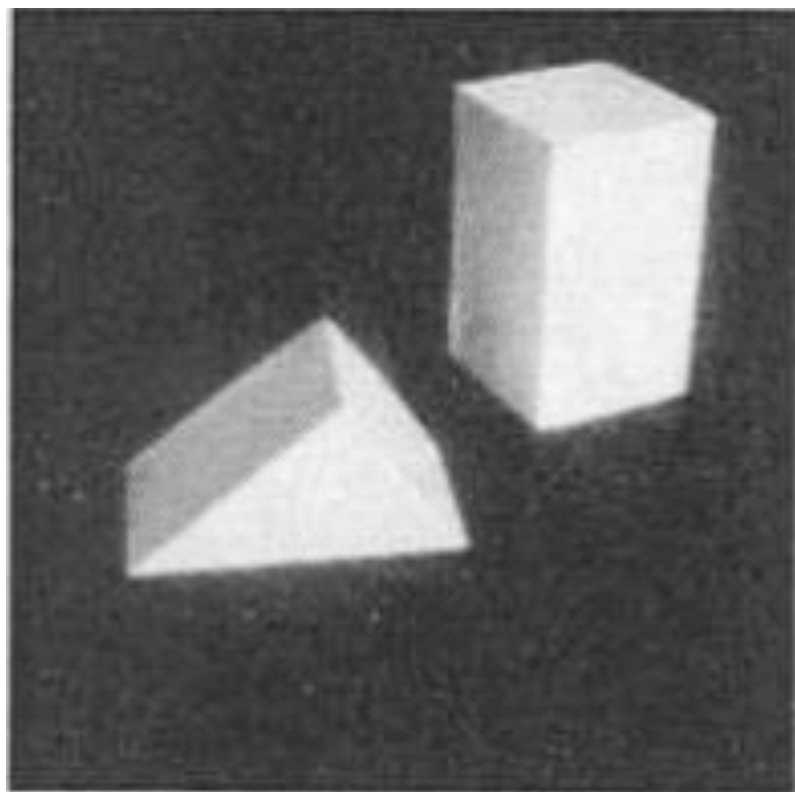
Scene understanding



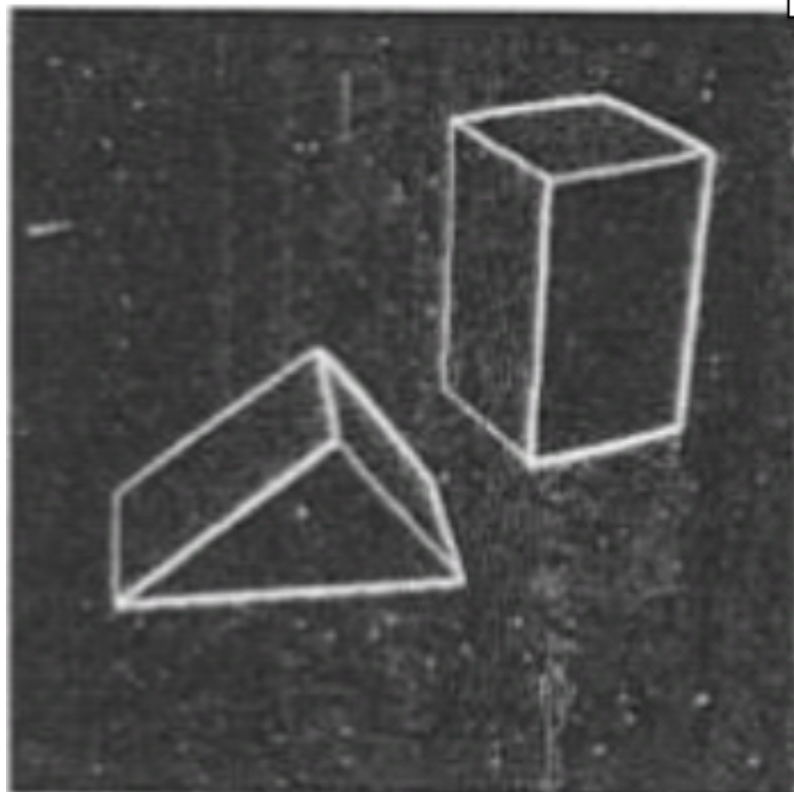
A bit of history...

So, let's make the problem simpler: Block's world

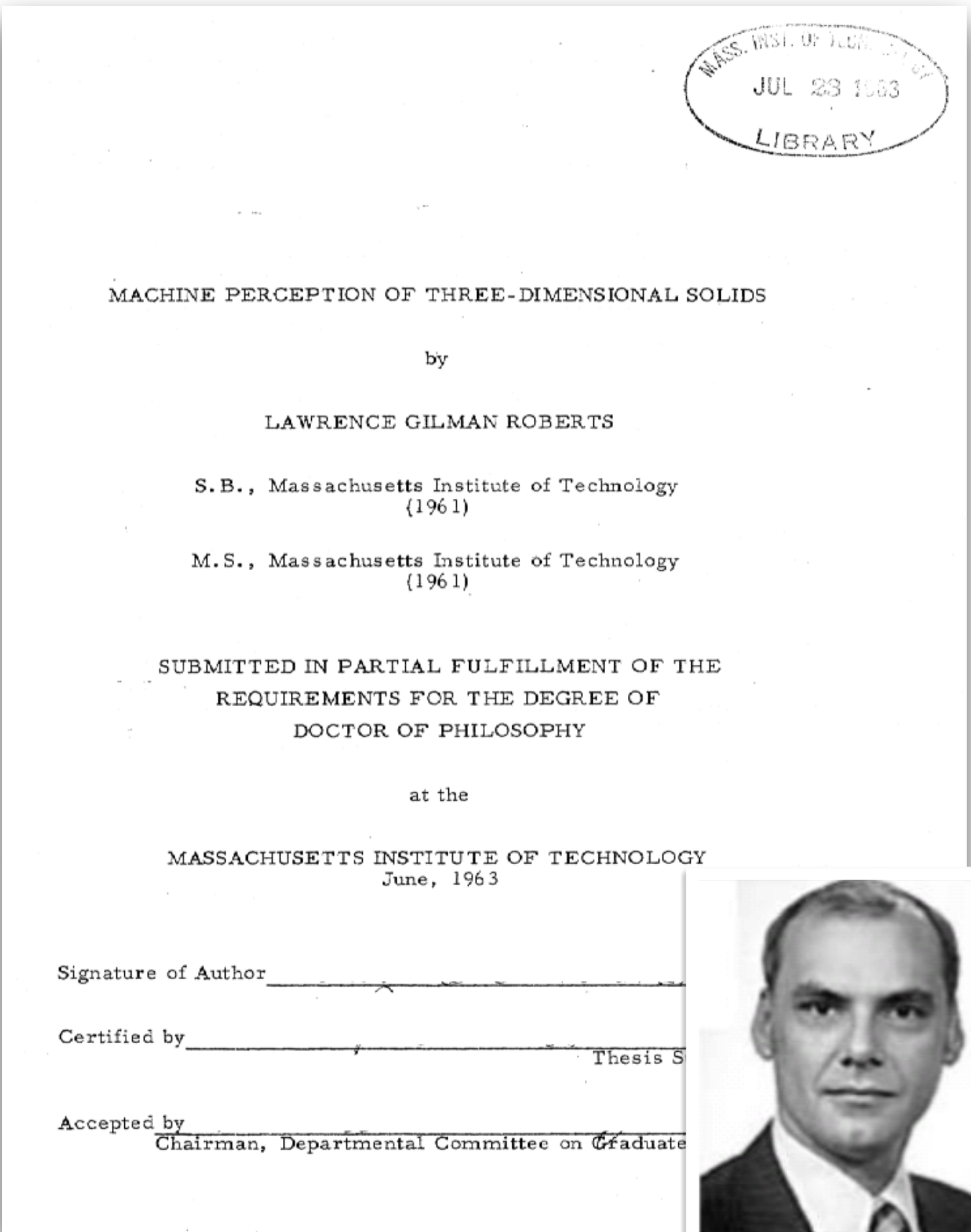
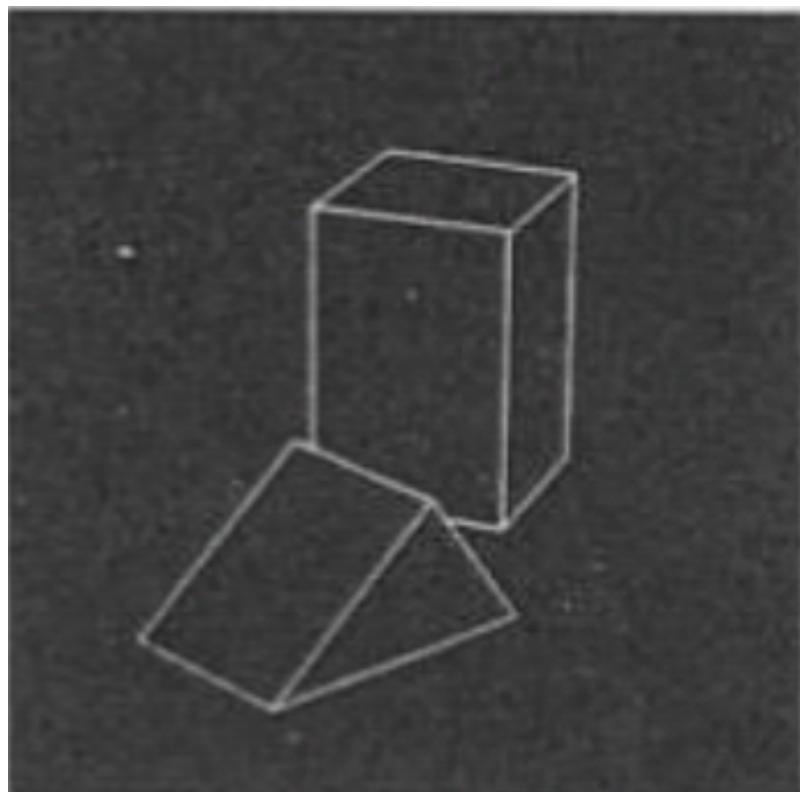
Input



Edges (2x2 gradient)



Reconstructed 3D scene
(new view point)



3D, compositional models

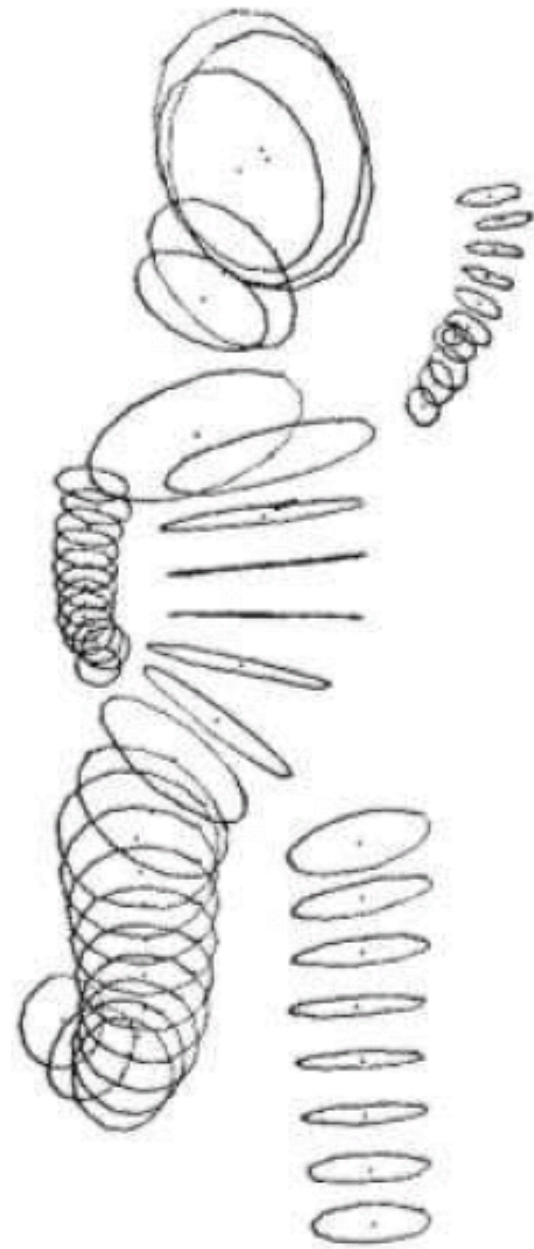
Binford and generalized cylinders



a)



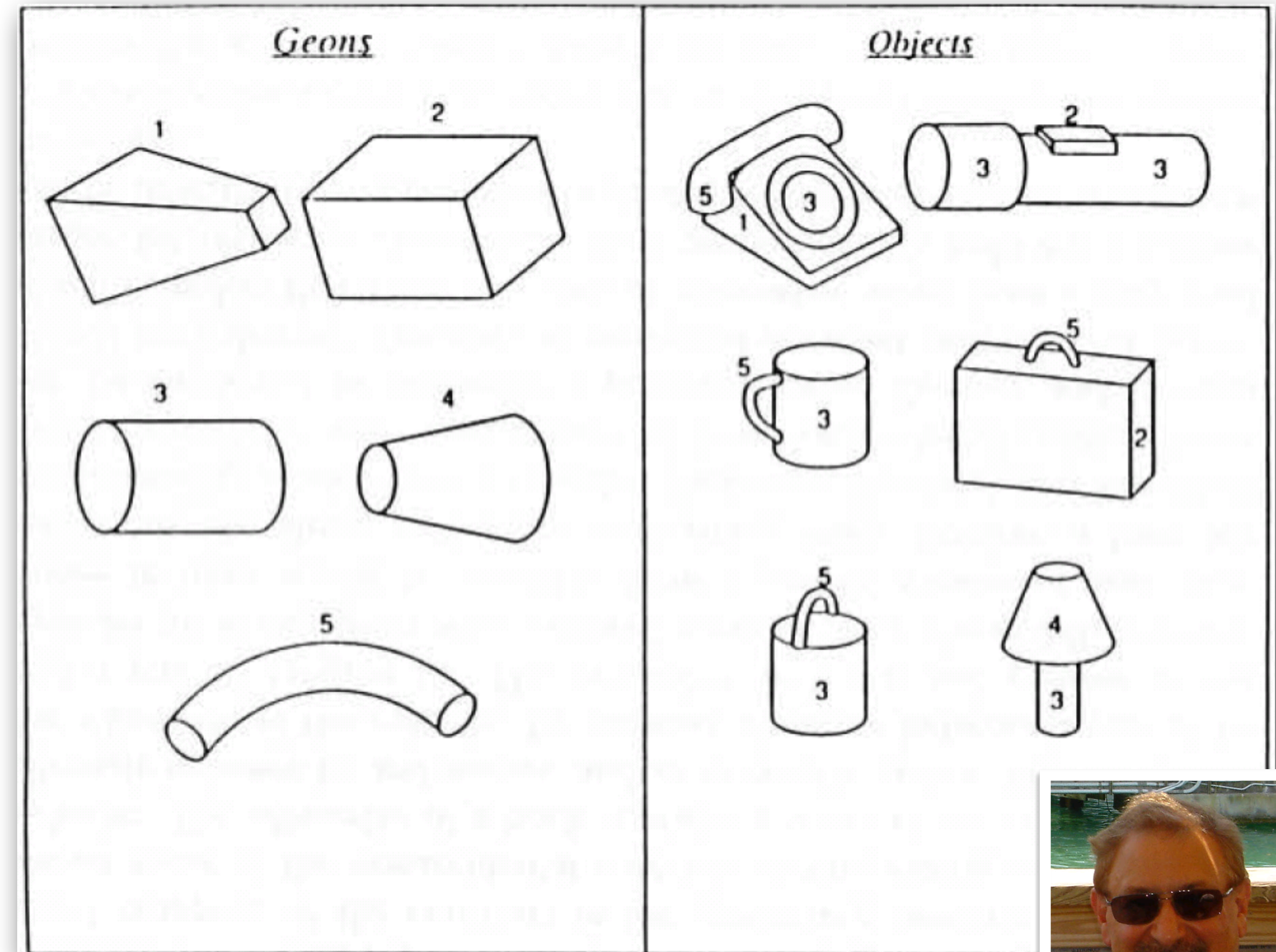
b)



c)

Object Recognition in the Geometric Era: a Retrospective. Joseph L. Mundy. 2006

Recognition by components

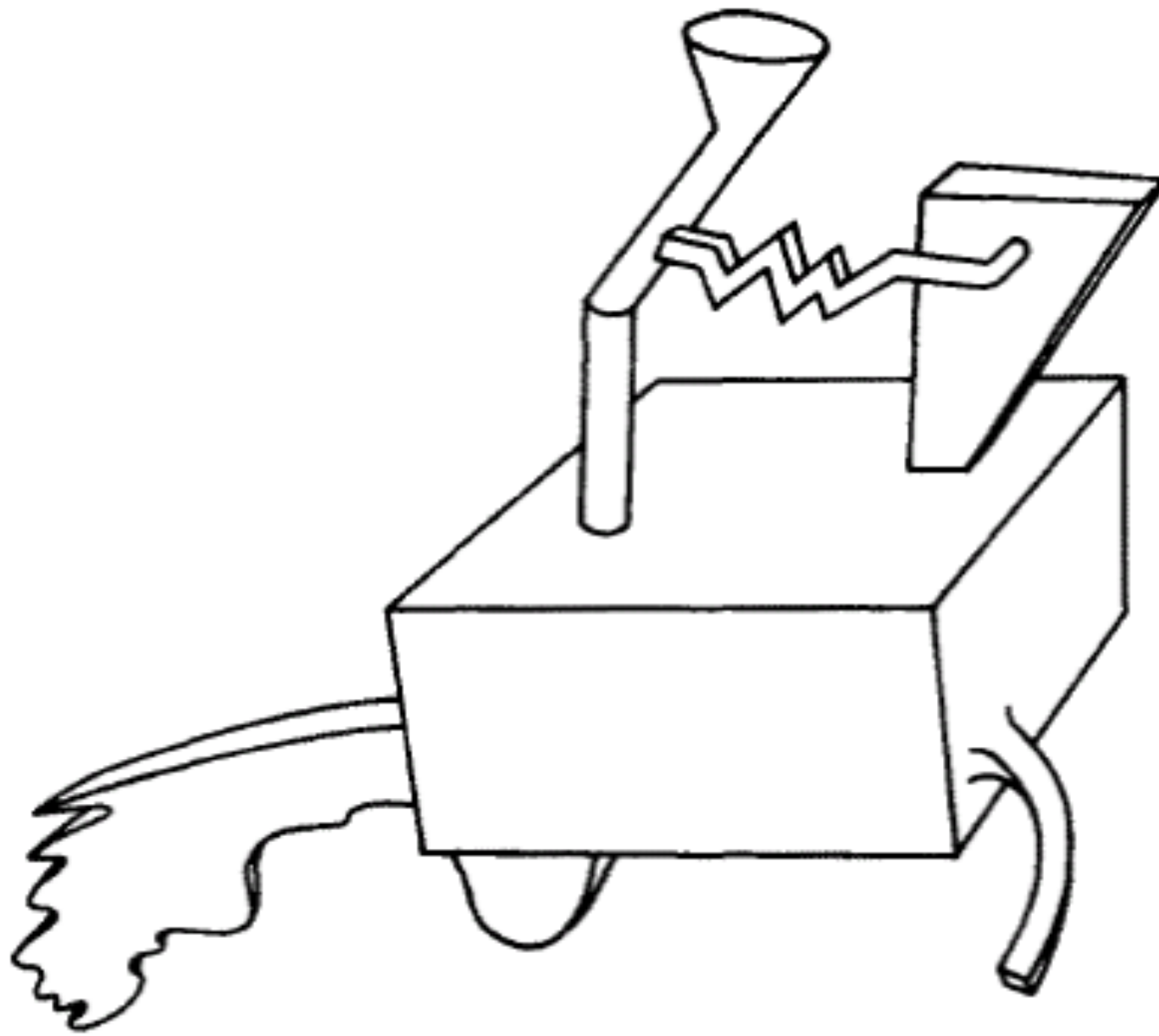


Recognition-by-Components: A Theory of Human Image Understanding. Psychological Review, 1987.



Irving Biederman

A do-it-yourself example



- 1) We know that this object is nothing we know
- 2) We can split this objects into parts that everybody will agree
- 3) We can see how it resembles something familiar: “a hot dog cart”

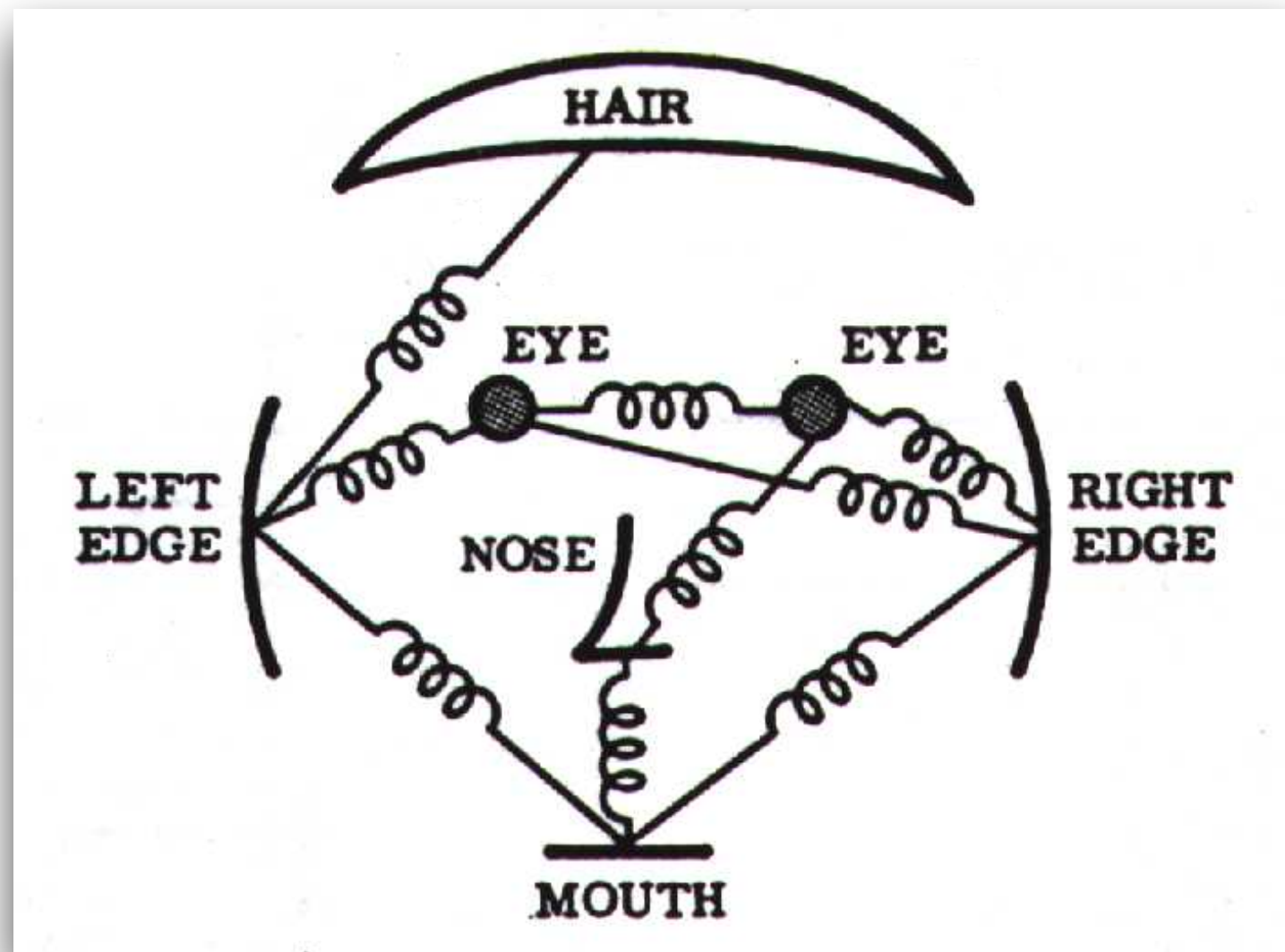
“The naive realism that emerges in descriptions of nonsense objects may be reflecting the workings of a representational system by which objects are identified.”

Irving Biederman

Recognition-by-Components: A Theory of Human Image Understanding.

Psychological Review, 1987.

Part based models



- Object as set of parts
 - Generative representation
- Model:
 - Relative locations between parts
 - Appearance of part
- Issues:
 - How to model location
 - How to represent appearance
 - Sparse or dense (pixels or regions)
 - How to handle occlusion/clutter

The Representation and Matching of Pictorial Structures

MARTIN A. FISCHLER AND ROBERT A. ELSCHLAGER

Abstract—The primary problem dealt with in this paper is the following. Given some description of a visual object, find that object in an actual photograph. Part of the solution to this problem is the specification of a descriptive scheme, and a metric on which to base the decision of “goodness” of matching or detection.

We offer a combined descriptive scheme and decision metric which is general, intuitively satisfying, and which has led to promising experimental results. We also present an algorithm which takes the above descriptions, together with a matrix representing the intensities of the actual photograph, and then finds the described object in the matrix. The algorithm uses a procedure similar to dynamic programming in order to cut down on the vast amount of computation otherwise necessary.

One desirable feature of the approach is its generality. A new programming system does not need to be written for every new description; instead, one just specifies descriptions in terms of a certain set of primitives and parameters.

There are many areas of application: scene analysis and description, map matching for navigation and guidance, optical tracking,

Manuscript received November 30, 1971; revised May 22, 1972, and August 21, 1972.

The authors are with the Lockheed Palo Alto Research Laboratory, Lockheed Missiles & Space Company, Inc., Palo Alto, Calif. 94304.

```
1234567890123456789012345678901234567890
1 1
2 1
3 1
4 1
5 1
6 1
7 1
8 1
9 1
10 1
11 1
12 1
13 1
14 1
15 1
16 1
17 1
18 1
19 1
20 1
21 1
22 1
23 1
24 1
25 1
26 1
27 1
28 1
29 1
30 1
31 1
32 1
33 1
34 1
35 1
```

Original picture.

```
1234567890123456789012345678901234567890
1 1
2 1
3 1
4 1
5 1
6 1
7 1
8 1
9 1
10 1
11 1
12 1
13 1
14 1
15 1
16 1
17 1
18 1
19 1
20 1
21 1
22 1
23 1
24 1
25 1
26 1
27 1
28 1
29 1
30 1
31 1
32 1
33 1
34 1
35 1
```

Noisy picture (sensed scene) as taken in experiment.

HAIR WAS LOCATED AT (6, 18)
L/EDGE WAS LOCATED AT (18, 10)
R/EDGE WAS LOCATED AT (18, 25)
L/EYE WAS LOCATED AT (17, 13)
R/EYE WAS LOCATED AT (17, 21)
NOSE WAS LOCATED AT (22, 15)
MOUTH WAS LOCATED AT (24, 17)

```
1234567890123456789012345678901234567890
1 1
2 1
3 1
4 1
5 1
6 1
7 1
8 1
9 1
10 1
11 1
12 1
13 1
14 1
15 1
16 1
17 1
18 1
19 1
20 1
21 1
22 1
23 1
24 1
25 1
26 1
27 1
28 1
29 1
30 1
31 1
32 1
33 1
34 1
35 1
```

L(EV)A for eye. (Density at a point is proportional to probability that an eye is present at that location.)

Scene models

Multiple levels of representation -- pixels > patches > regions > subimages > objects

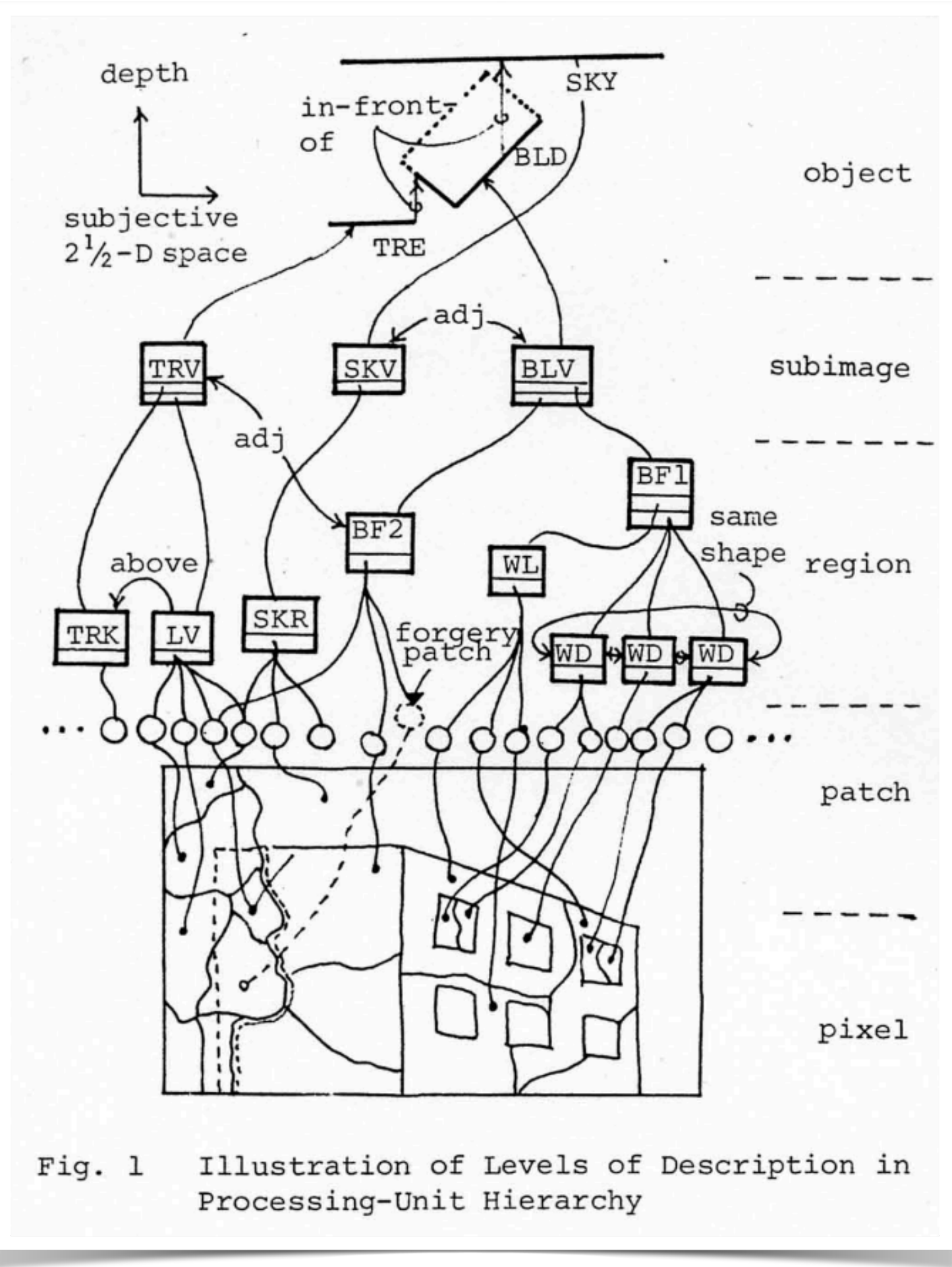


Fig. 1 Illustration of Levels of Description in Processing-Unit Hierarchy

ABSTRACT

This paper overviews and discusses model representations and control structures in image understanding. Hierarchies are observed in the levels of description used in image understanding along a few dimensions: processing unit, detail, composition and scene/view distinction. Emphasis is placed on the importance of explicitly handling the hierarchies both in representing knowledge and in using it. A scheme of "knowledge block" representation which is structured along the processing-unit hierarchy is also presented.

I. INTRODUCTION

Image Understanding System(IUS) constructs a description of the scene being viewed from an array of image sensory data: intensity, color, and sometimes range data. Image understanding is best characterized by description, whereas pattern recognition by classification, and image processing by image output. The level and scope of the goal description depend on the task given to the IUS: whether it is interpretation, object detection, change detection, image matching, etc. It may appear that the discussion in this paper will take usually the flavor of scene interpretation from a monocular intensity image.

Observing that there are hierarchies of levels of description along a few dimensions, this paper overviews and discusses model representations and control structures in image understanding. Emphasis is placed on the importance of explicitly handling the hierarchies both in representing knowledge about scenes and in using it, especially processing-unit hierarchy and scene/view domain distinction.

In the next section, the levels of description are identified. Then section III gives an overview and discussion on object-model representations, together with presentation of our knowledge block representation scheme. Section IV deals with the problems of control structure, and finally the role of low-level processing is discussed in section V.

II. LEVELS OF DESCRIPTION IN IMAGE UNDERSTANDING

Descriptions are not only the goal constructs, but also the media through which various components of an IUS communicate in the course of understanding the image. There are a few orthogonal dimensions.

a) Processing-unit Hierarchy

This is a hierarchy in the levels of units used in processing. Let us identify five levels for the moment. For a region-based IUS, they are pixel (an image point), patch(a group of contiguous pixels having similar pixel properties), region(a meaningful group of patches corresponding to a surface of an object), subimage(a part of an image

corresponding to an object or a set of objects), and object(an object as a real entity). For a line-based IUS, the level of patch can be replaced by line segment, region by line, and subimage by a set of lines corresponding to an object, Fig. 1 illustrates these levels for a region-based IUS.

Akin & Reddy(1976) observed that six levels are used when human subjects understand the contents of an image through verbal conversation: scene, cluster, object, region, segment, and intensity. The number of levels is not very significant. These levels as well as those in Fig. 1 depend on the units on which different levels of processing are performed and for whose description different vocabularies are used. Processing in the pixel-to-patch level is often called as low-level processing. The region-to-subimage level is high level in the picture processing domain. It clearly needs to deal with semantics which stem from the highest, object level. The patch-to-region level might be called as intermediate.

b) View Domain / Scene Domain Distinction

The point to be noted here is the clear disparity existing between view-domain and scene-domain descriptions; in Fig. 1, the lower four levels are in the view domain and the upper one in scene domain. The need for this distinction was argued for first and most effectively by Clowes(1971). He used the term "picture domain" in place of "view domain". But the latter is used in this paper to mean the domain of observable facts by viewing the scene in either intensity or range data. The importance of this distinction is readily understood by thinking that, for example, the actual meaning of "adjacency" in the view-domain description is fully understood only after the relation is interpreted in the scene-domain description. Note that the scene-domain descriptions are not necessarily in a metrical 3-D coordinate space; e.g., Waltz's labels of edge is a symbolic system to represent the edge types in the 3-D space, or even a gross subjective space will suffice.

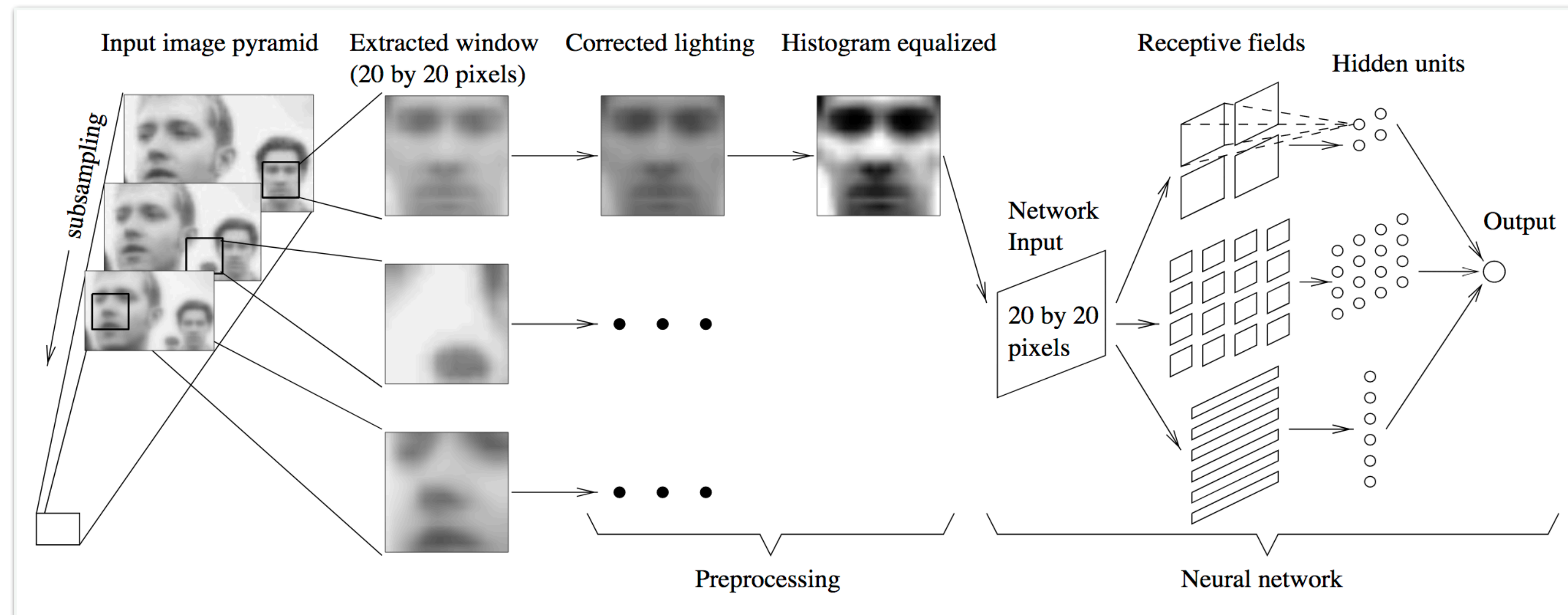
c) Detail Hierarchy and Composition Hierarchy

The detail hierarchy is along preciseness of description. It can exist in both the view and the scene domains. Section 5.2 presents examples in the view domain. An example in the scene domain is the description of overall/detail shape of an object, which is found in section 3.2b). The composition (or part-of) hierarchy represents part/whole relationships in the scene domain.

The processing-unit hierarchy actually contains somewhat both aspects of the detail and composition hierarchies in the sense that the low-level entities are parts and details of an upper-level entity. Unfortunately this revealed hierarchy does not directly correspond to the hierarchies which naturally exist in the scene domain. This fact makes image understanding difficult, and it is why the models often need to represent the natural hierarchies

Neural Network-Based Face Detector

Train a set of multilayer perceptrons and arbitrate a decision among all outputs



Rowley, Baluja, and Kanade: Neural Network-Based Face Detection (PAMI, January 1998)

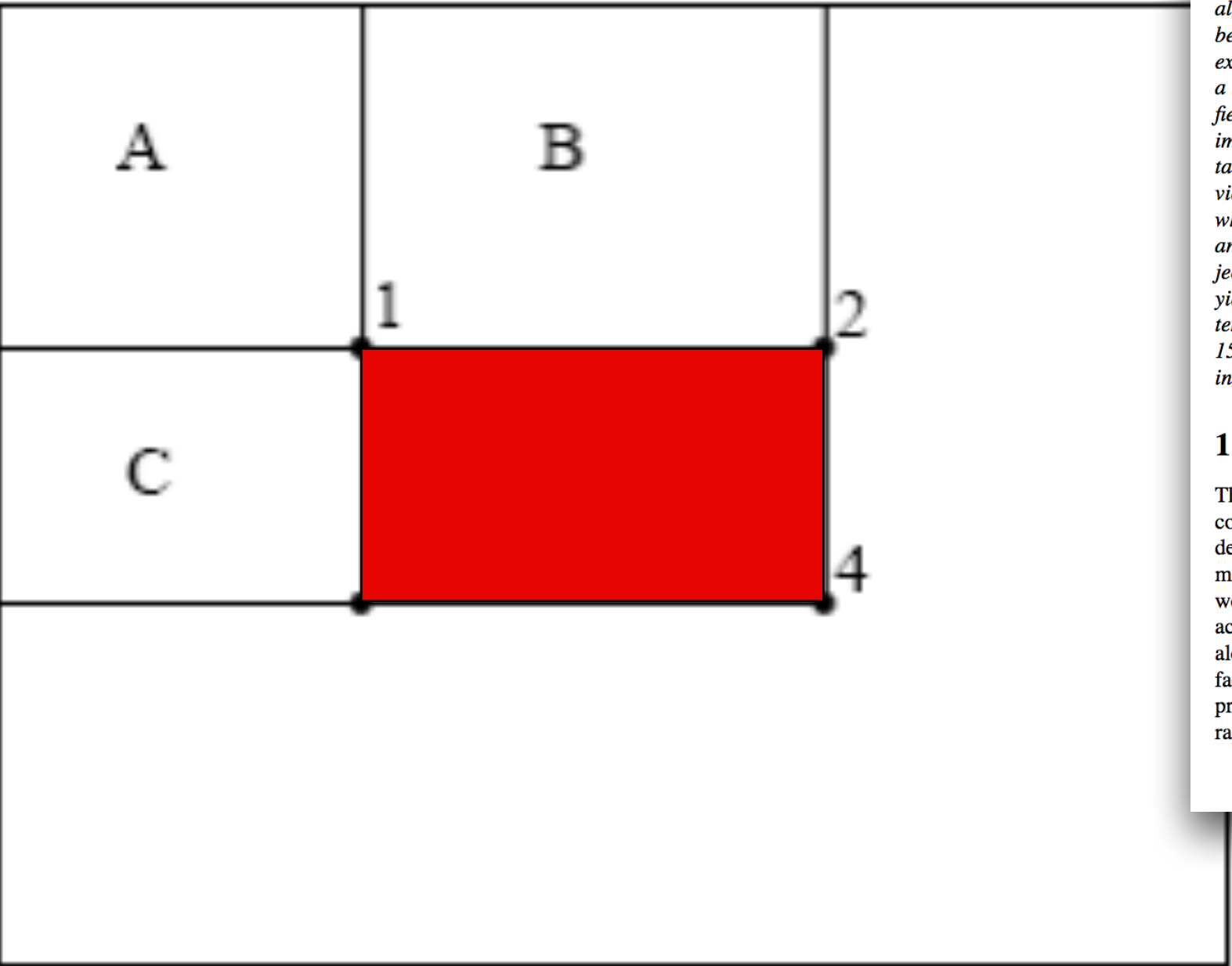
Viola-Jones Face Detector

Haar filters, integral image and boosting

Viola and Jones, ICCV 2001



Integral image



The average intensity in the block is computed with four sums independently of the block size.

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.

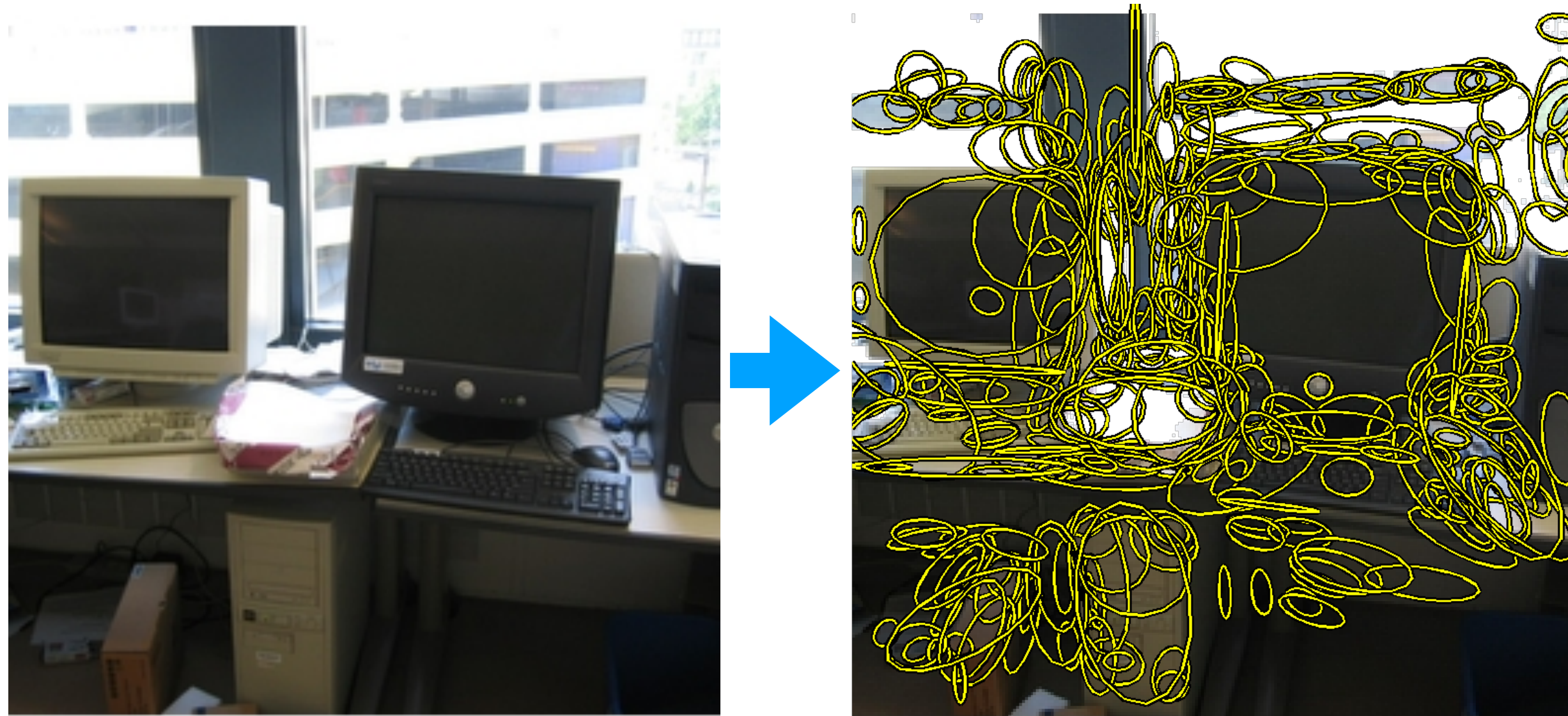
There are three main contributions of our object detection framework. We will introduce each of these ideas briefly below and then describe them in detail in subsequent sections.

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 Papa-georgiou 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 sub-window 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

1

Bag of words models

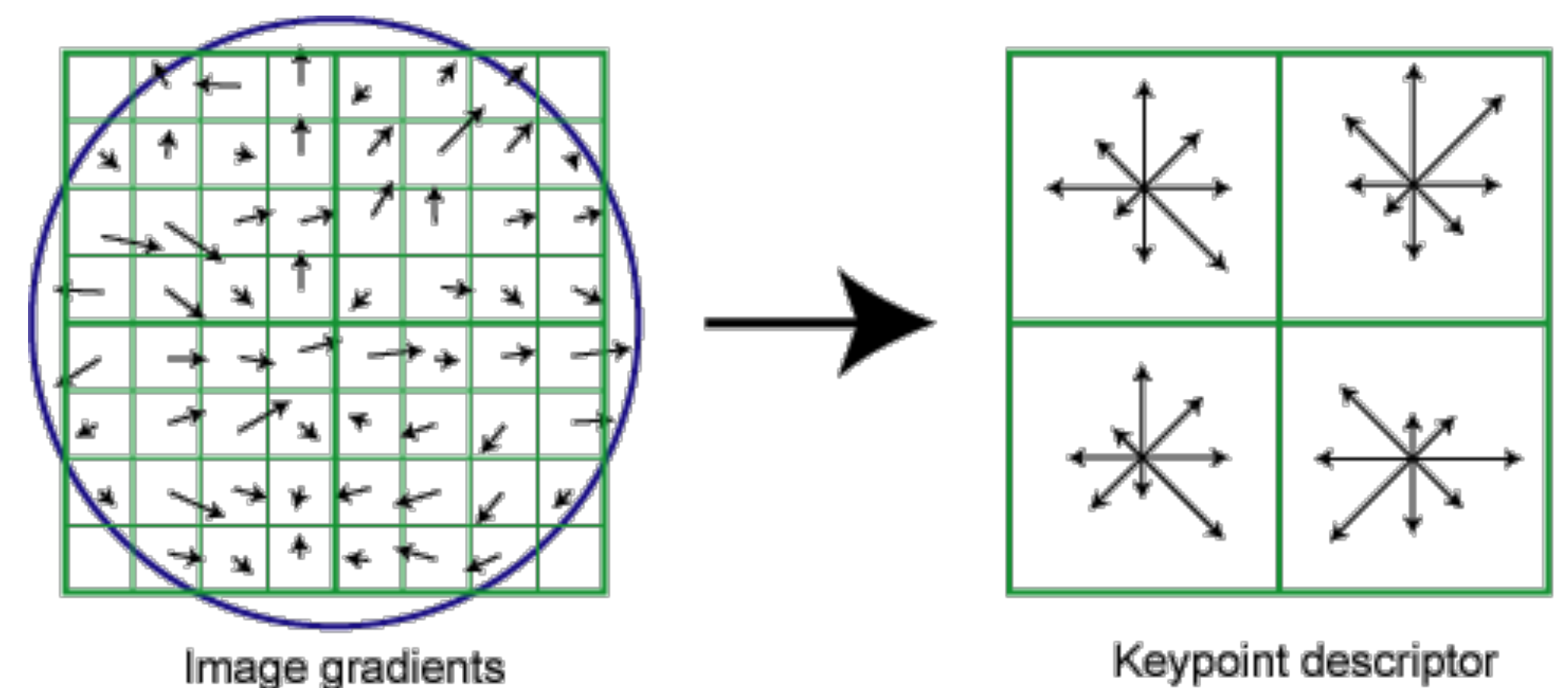


Images represented as *affine covariant regions*:
Harris affine invariant regions (corners & edges)

Maximally stable extremal regions
(segmentation)

Csurka, Dance, Fan, Willamowski, and Bray 2004
Sivic, Russell, Freeman, Zisserman, ICCV 2005

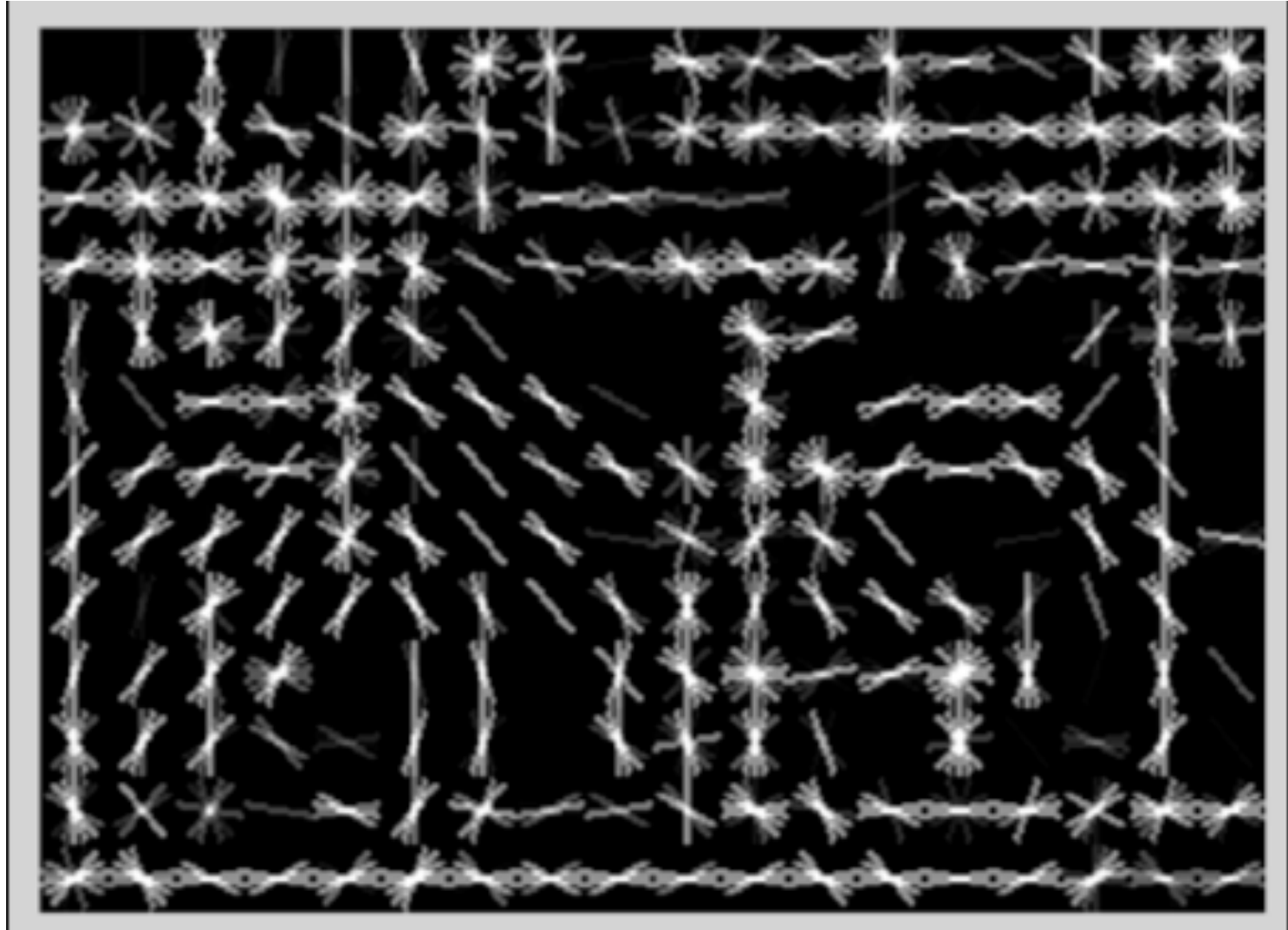
- SIFT: Scale Invariant Feature Transform
- Normalized histogram of orientation energy in each affinely adapted region (128-dim.)



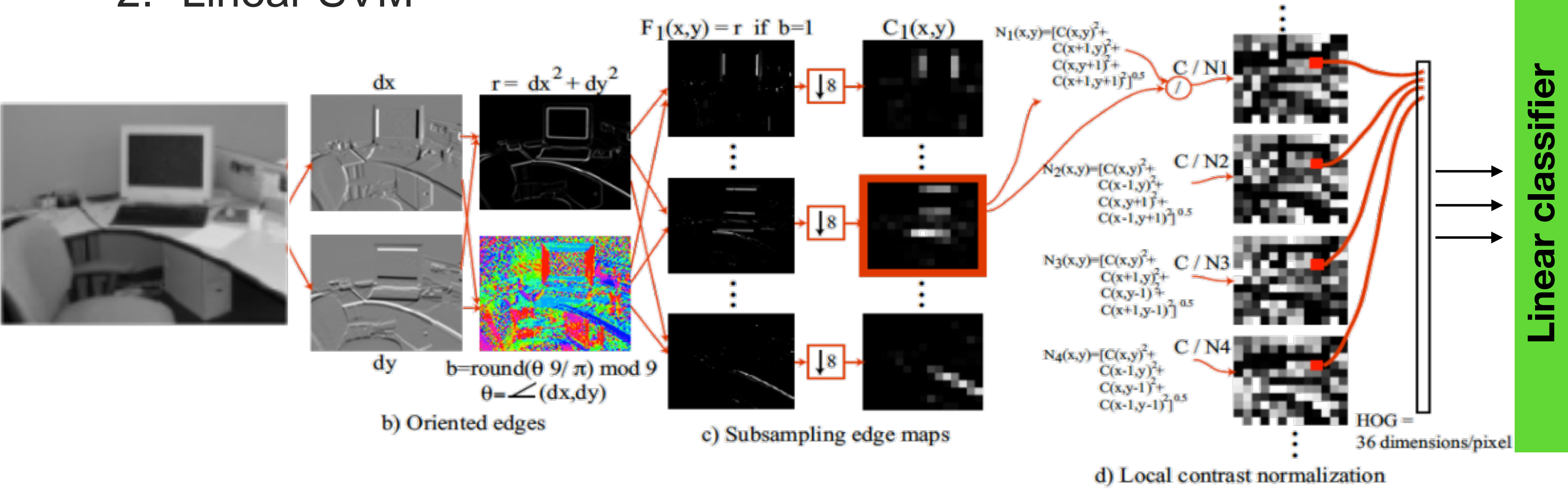
D. Lowe, IJCV 2004

Histograms of oriented gradients (HOG)

<https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf> 2005



1. Bin gradients from 8x8 pixel neighborhoods into 9 orientations
2. Linear SVM



Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs
INRIA Rhône-Alpes, 655 avenue de l'Europe, Montbonnot 38334, France
{Navneet.Dalal,Bill.Triggs}@inrialpes.fr, <http://lear.inrialpes.fr>

Abstract

We study the question of feature sets for robust visual object recognition, adopting linear SVM based human detection as a test case. After reviewing existing edge and gradient based descriptors, we show experimentally that grids of Histograms of Oriented Gradient (HOG) descriptors significantly outperform existing feature sets for human detection. We study the influence of each stage of the computation on performance, concluding that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good results. The new approach gives near-perfect separation on the original MIT pedestrian database, so we introduce a more challenging dataset containing over 1800 annotated human images with a large range of pose variations and backgrounds.

1 Introduction

Detecting humans in images is a challenging task owing to their variable appearance and the wide range of poses that they can adopt. The first need is a robust feature set that allows the human form to be discriminated cleanly, even in cluttered backgrounds under difficult illumination. We study the issue of feature sets for human detection, showing that locally normalized Histogram of Oriented Gradient (HOG) descriptors provide excellent performance relative to other existing feature sets including wavelets [17,22]. The proposed descriptors are reminiscent of edge orientation histograms [4,5], SIFT descriptors [12] and shape contexts [1], but they are computed on a dense grid of uniformly spaced cells and they use overlapping local contrast normalizations for improved performance. We make a detailed study of the effects of various implementation choices on detector performance, taking "pedestrian detection" (the detection of mostly visible people in more or less upright poses) as a test case. For simplicity and speed, we use linear SVM as a baseline classifier throughout the study. The new detectors give essentially perfect results on the MIT pedestrian test set [18,17], so we have created a more challenging set containing over 1800 pedestrian images with a large range of poses and backgrounds. Ongoing work suggests that our feature set performs equally well for other shape-based object classes.

2 Previous Work

There is an extensive literature on object detection, but here we mention just a few relevant papers on human detection [18,17,22,16,20]. See [6] for a survey. Papageorgiou *et al* [18] describe a pedestrian detector based on a polynomial SVM using rectified Haar wavelets as input descriptors, with a parts (subwindow) based variant in [17]. Depoortere *et al* give an optimized version of this [2]. Gavrilu & Philomen [8] take a more direct approach, extracting edge images and matching them to a set of learned exemplars using chamfer distance. This has been used in a practical real-time pedestrian detection system [7]. Viola *et al* [22] build an efficient moving person detector, using AdaBoost to train a chain of progressively more complex region rejection rules based on Haar-like wavelets and space-time differences. Ronfard *et al* [19] build an articulated body detector by incorporating SVM based limb classifiers over 1st and 2nd order Gaussian filters in a dynamic programming framework similar to those of Felzenszwalb & Huttenlocher [3] and Ioffe & Forsyth [9]. Mikolajczyk *et al* [16] use combinations of orientation-position histograms with binary-thresholded gradient magnitudes to build a parts based method containing detectors for faces, heads, and front and side profiles of upper and lower body parts. In contrast, our detector uses a simpler architecture with a single detection window, but appears to give significantly higher performance on pedestrian images.

3 Overview of the Method

This section gives an overview of our feature extraction chain, which is summarized in fig. 1. Implementation details are postponed until §6. The method is based on evaluating well-normalized local histograms of image gradient orientations in a dense grid. Similar features have seen increasing use over the past decade [4,5,12,15]. The basic idea is that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or

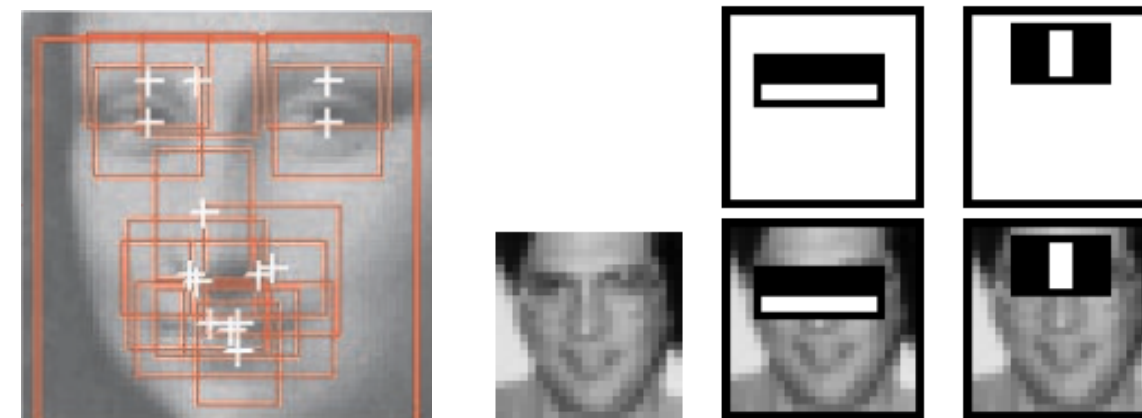
Families of recognition algorithms

Bag of words models



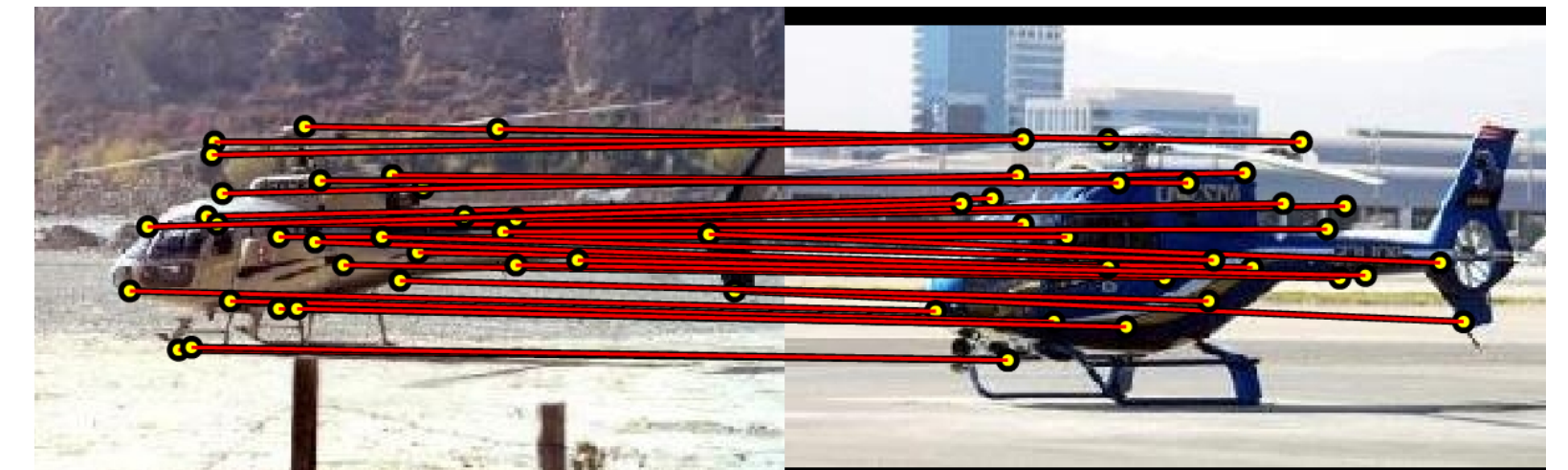
Csurka, Dance, Fan, Willamowski, and Bray 2004
Sivic, Russell, Freeman, Zisserman, ICCV 2005

Voting models



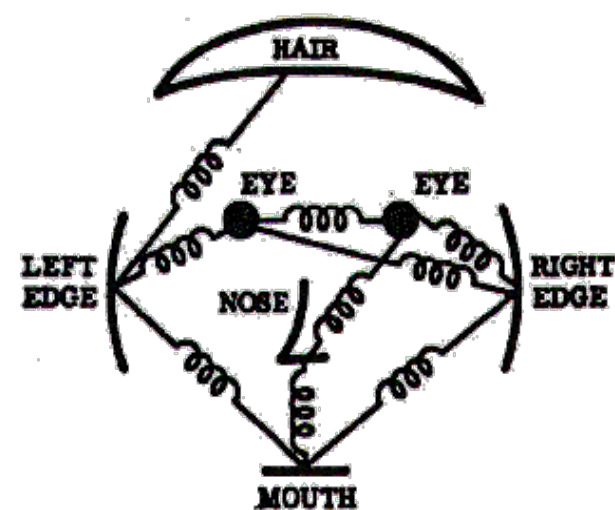
Viola and Jones, ICCV 2001
Heisele, Poggio, et. al., NIPS 01
Schneiderman, Kanade 2004
Vidal-Naquet, Ullman 2003

Shape matching Deformable models



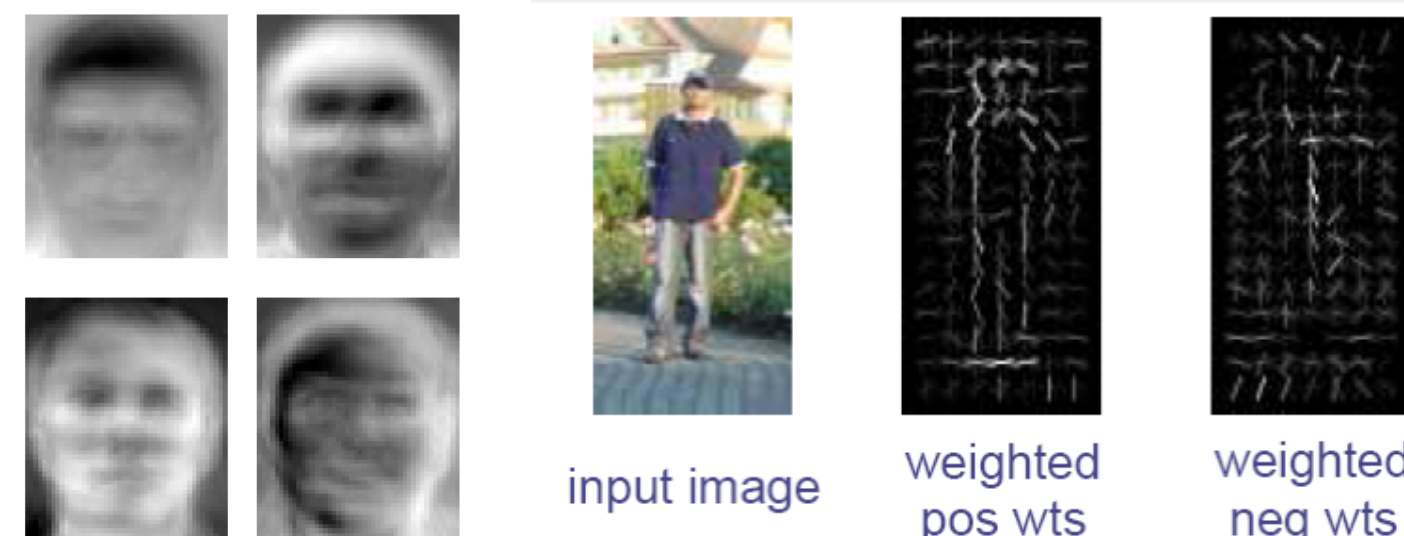
Berg, Berg, Malik, 2005
Cootes, Edwards, Taylor, 2001

Constellation models



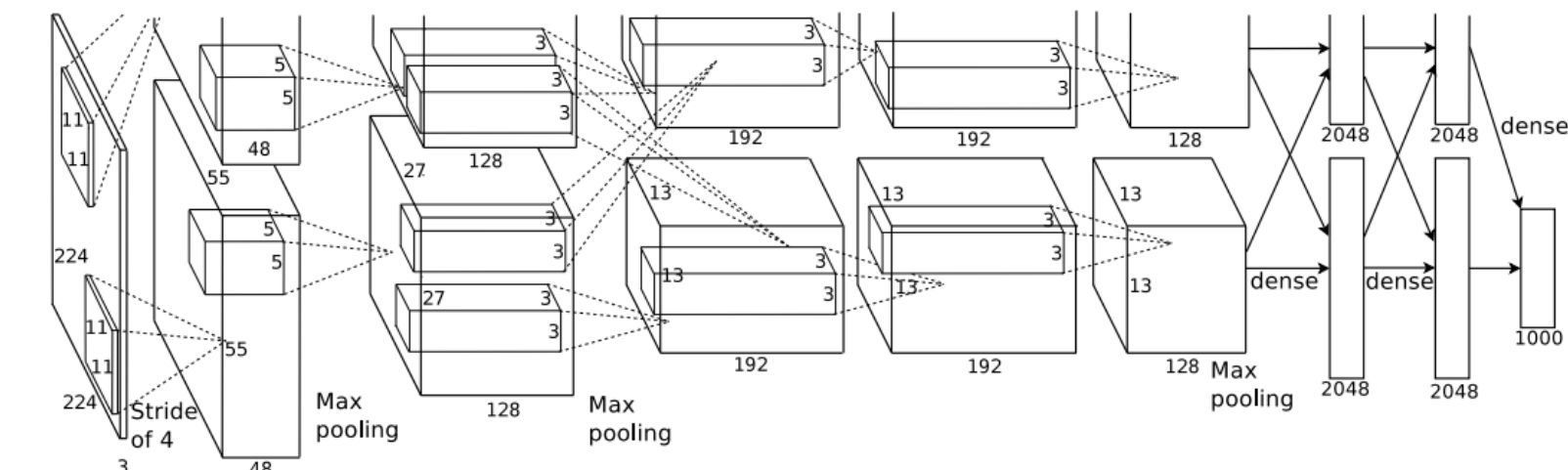
Fischler and Elschlager, 1973
Burl, Leung, and Perona, 1995
Weber, Welling, and Perona, 2000
Fergus, Perona, & Zisserman, CVPR 2003

Rigid template models



Sirovich and Kirby 1987
Turk, Pentland, 1991
Dalal & Triggs, 2006

Neural networks



Le Cun et al, 98






car



ImageNet classification and Neural nets

IMGENET


14,197,122 images, 21841 synsets indexed

[Explore](#) [Download](#) [Challenges](#) [Publications](#) [CoolStuff](#) [About](#)

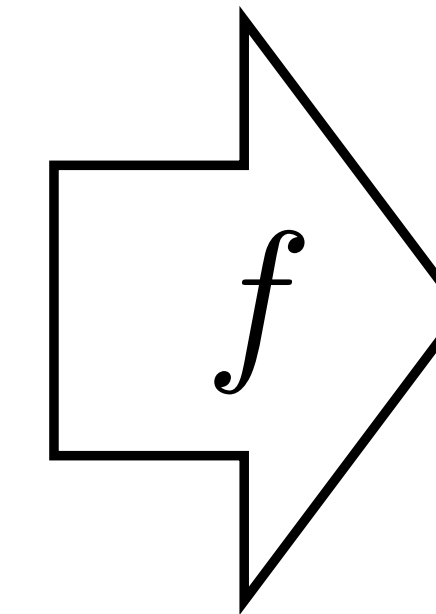
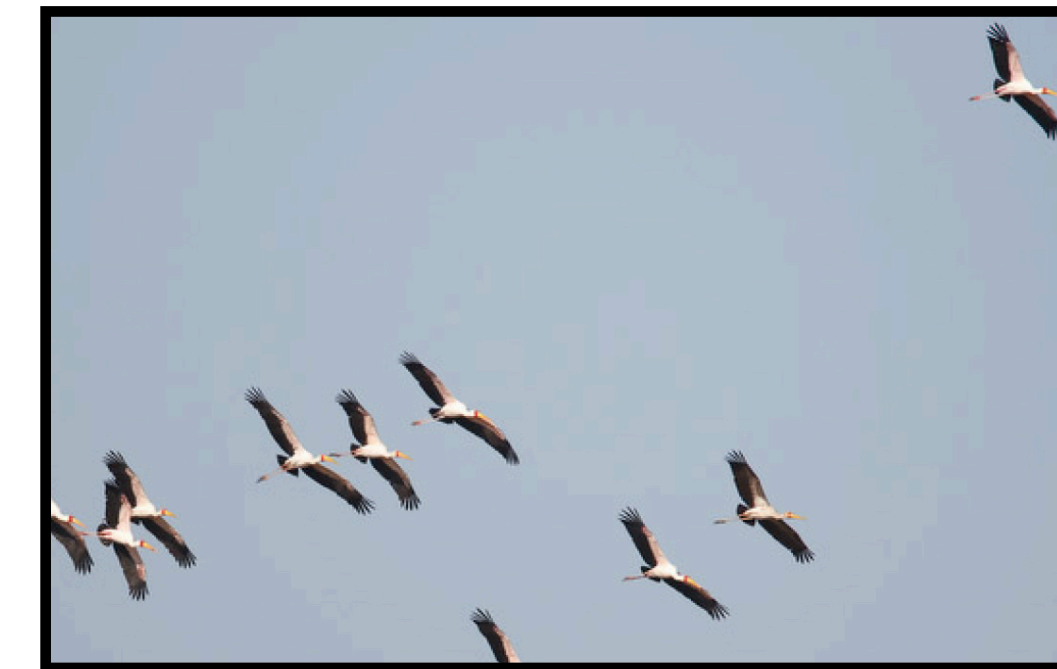
Not logged in. [Login](#) | [Signup](#)

ImageNet is an image database organized according to the **WordNet** hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. Currently we have an average of over five hundred images per node. We hope ImageNet will become a useful resource for researchers, educators, students and all of you who share our passion for pictures.

[Click here](#) to learn more about ImageNet, [Click here](#) to join the ImageNet mailing list.



What do these images have in common? *Find out!*



“Birds”

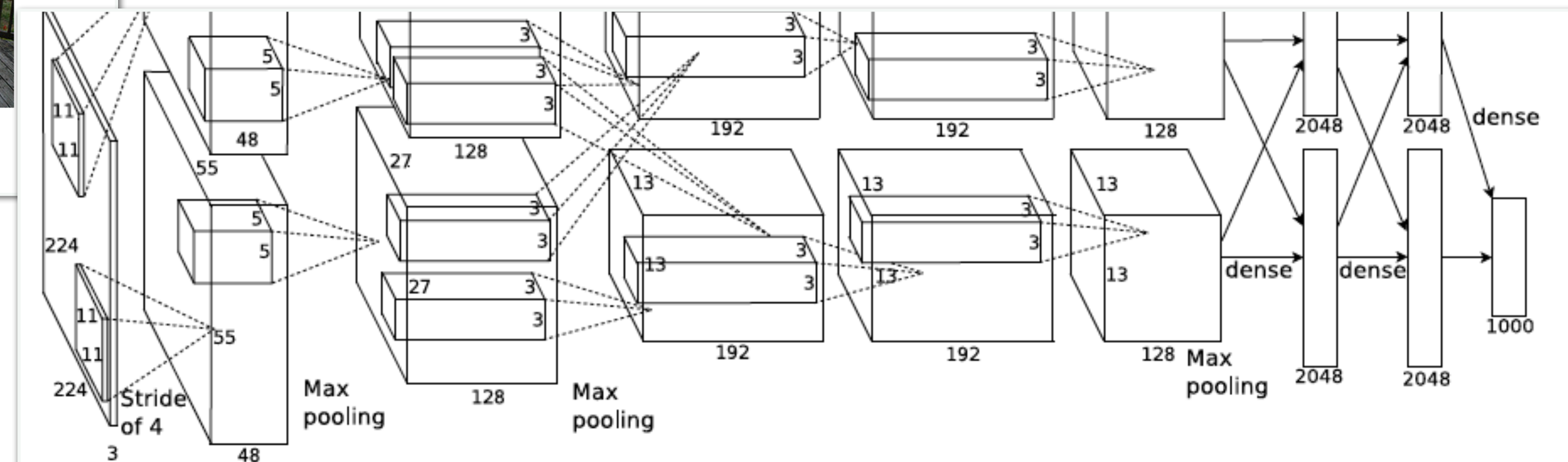


Image Classification

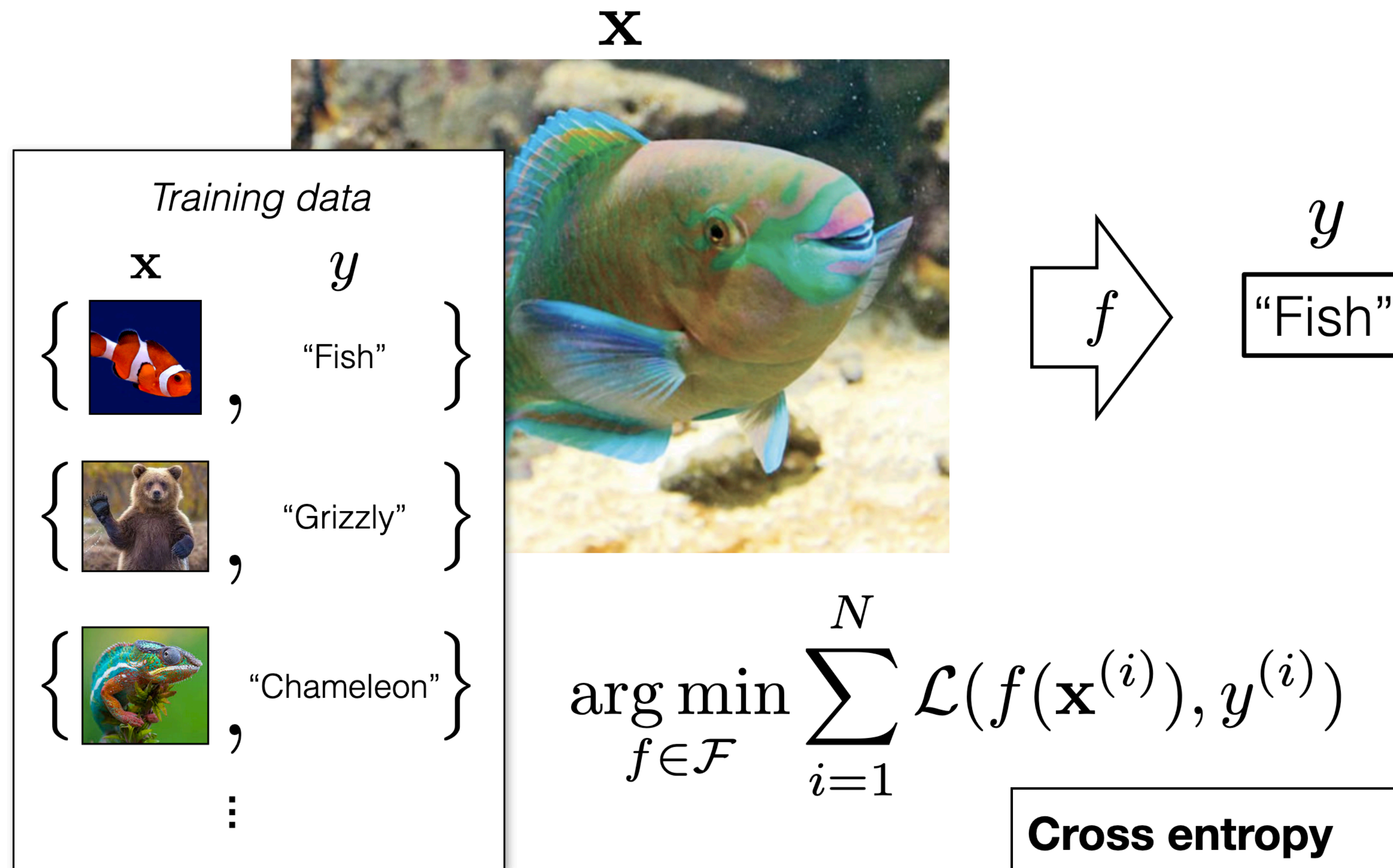
Is object class c present anywhere in the image x ?

Is there a **car** in this image?



→ Yes

Image Classification: formulation



Cross entropy

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = H(\mathbf{y}, \hat{\mathbf{y}}) = - \sum_{k=1}^K y_k \log \hat{y}_k$$

Image Classification: evaluation

Classification performance (Top-n)

Predicted class

true class

$$\text{TOP-1} = \frac{100}{T} \sum_{t=1}^T \mathbb{1}(\hat{y}^{(t)} = y^{(t)})$$

Percentage of times that true class is correctly identified.

Image Classification: evaluation

Classification performance (Top-n)

$$\text{TOP-1} = \frac{100}{T} \sum_{t=1}^T \mathbb{1}(\hat{y}^{(t)} = y^{(t)})$$

Predicted class $\hat{y}^{(t)}$ and true class $y^{(t)}$ are indicated by red arrows.

Percentage of times that true class is correctly identified.

Confusion matrix

$$C_{ij} = 100 \frac{\sum_{t=1}^T \mathbb{1}(\hat{y}^{(t)} = j) \mathbb{1}(y^{(t)} = i)}{\sum_{t=1}^T \mathbb{1}(y^{(t)} = i)}$$

Predicted class $\hat{y}^{(t)} = j$ and true class $y^{(t)} = i$ are indicated by red arrows.

where $C_{i,j}$ measures the percentage of times that true class i is classified as class j .

		Predicted class			
		Cat	Car	Dog	
True class	Cat	80%	5%	15%	$\rightarrow \sum = 100\%$
	Car	2%	95%	3%	$\rightarrow \sum = 100\%$
	Dog	4%	0%	96%	$\rightarrow \sum = 100\%$

Image Classification: shortcomings

Ground truth-annotation: Indicate which images contain a car

“Clap” if you see a car

Ready?











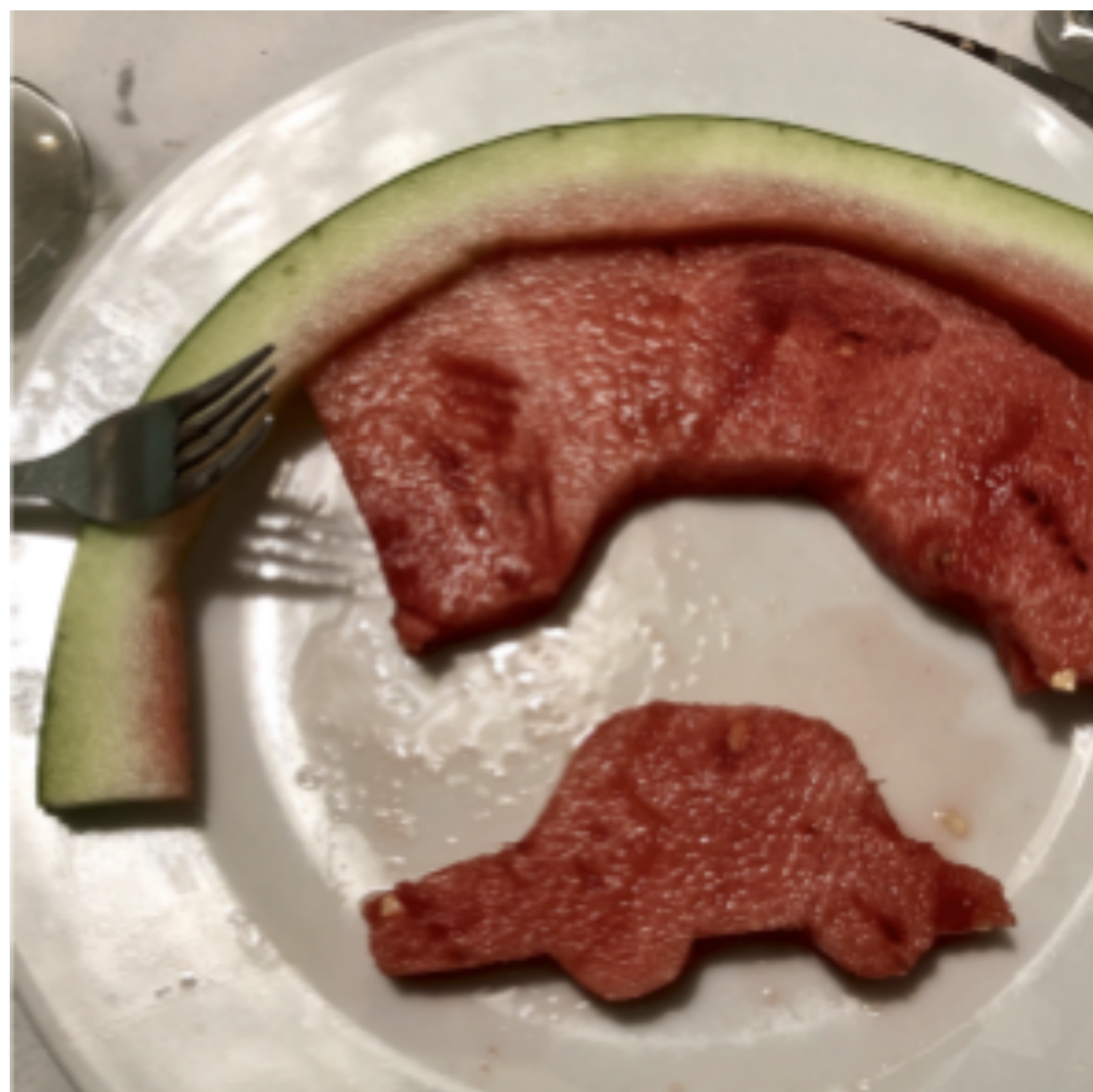


Image Classification: shortcomings

What is a car?



The data represents the formulation of the problem.

There is ambiguity even for very familiar concepts

Is there a fruit in this picture?



A pepper is a fruit according to botanics, and it is a vegetable according to the culinary classification.

Which level of categorization is the right one?



If you are thinking in buying a car, you might want to be a bit more specific about your categorization.

Entry-level categories

(Jolicoeur, Gluck, Kosslyn 1984)

- Typical member of a basic-level category are categorized at the expected level
- Atypical members tend to be classified at a subordinate level.



A bird



An ostrich

Class experiment

Class experiment

Experiment 1: draw a horse (the entire body, not just the head) in a white piece of paper.

Do not look at your neighbor! You already know how a horse looks like... no need to cheat.

Class experiment

Experiment 2: draw a horse (the entire body, not just the head) but this time chose a viewpoint as weird as possible.

View typicality



Despite we can categorize all three pictures as being views of a horse, the three pictures do not look as being equally typical views of horses. And they do not seem to be recognizable with the same easiness.

Canonical Perspective

Examples of canonical perspective:

Experiment (Palmer, Rosch & Chase 81): participants are shown views of an object and are asked to rate “how much each one looked like the objects they depict” (scale; 1=very much like, 7=very unlike)

In a recognition task, reaction time correlated with the ratings.



HORSE



PIANO



TEAPOT



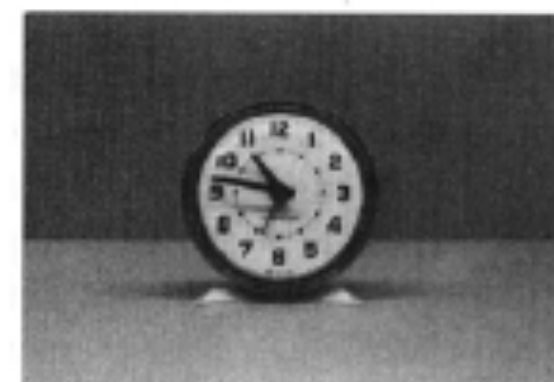
CAR



CHAIR



CAMERA



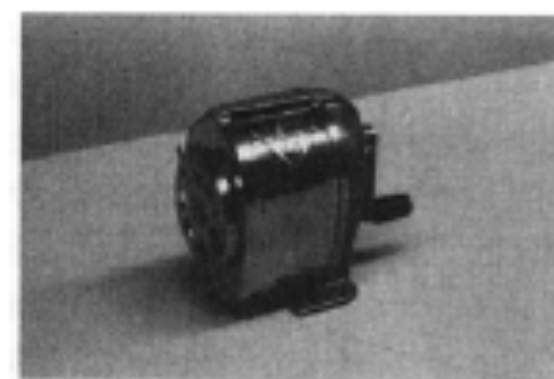
CLOCK



TELEPHONE



HOUSE



PENCIL SHARPENER



SHOE

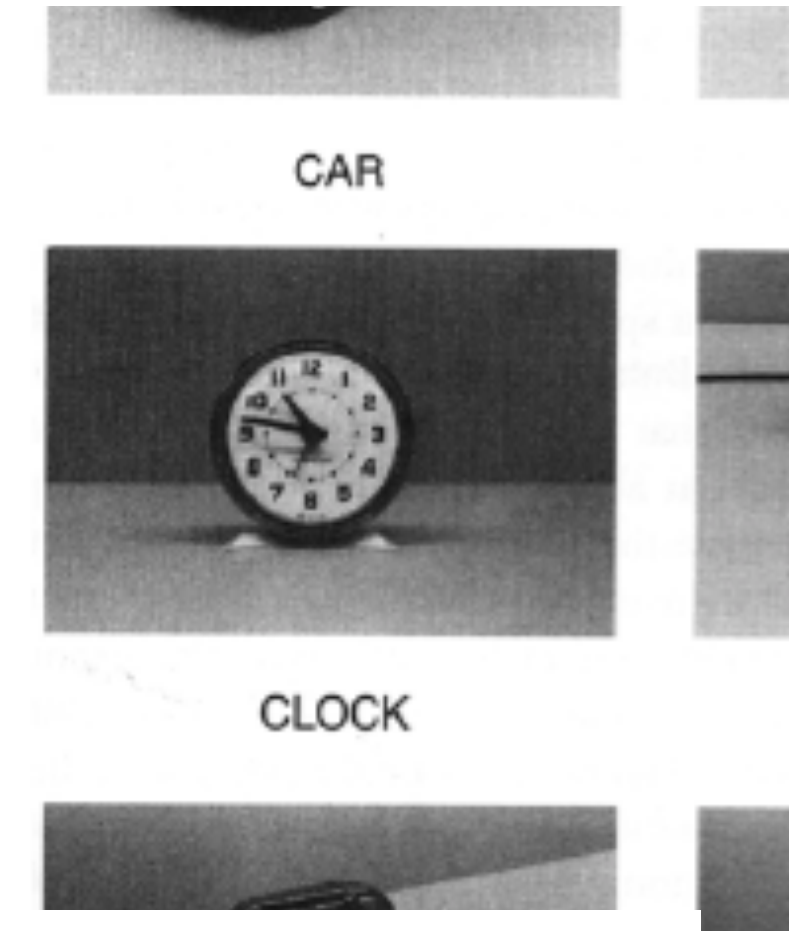


IRON

From *Vision Science*, Palmer

Canonical Viewpoint


Clocks are preferred as purely frontal




Google [Advanced Image Search](#) [Preferences](#)
[Moderate SafeSearch is on](#)

Images Showing: Results 1 - 18 of about 38,300,000 for


Related searches: [cartoon clock](#) [clock clipart](#) [alarm clock](#) [clock face](#)




clock character
359 x 344 - 4k - gif
school.discoveryeducation.com




Wind-up alarm clocks have been
...
346 x 510 - 22k - jpg
electronics.howstuffworks.com



Artistic Clock And Wall Clock
360 x 360 - 18k - jpg
www.global-b2b-network.com



... mechanical clock
screensaver.
640 x 480 - 53k - jpg
davinciautomata.wordpress.com



If it is 3 o'clock and we add 5 ...
305 x 319 - 4k - gif
www-math.cudenver.edu
[[More from](#)
www-math.cudenver.edu]

mug

Search

SafeSearch moderate

About 10,100,000 results (0.09 seconds)

Advanced search

59¢ Logo Coffee Mugs

www.DiscountMugs.com

Lead Free & Dishwasher Safe. Save 40-50%. No Catch. Factory Direct !

Custom Mugs On Sale

www.Vistaprint.com

Order Now & Save 50% On Custom Mugs No Minimums. Upload Photos & Logos.


Promotional Mugs from 69¢

www.4imprint.com/Mugs


Huge Selection of Styles Colors- Buy 72 Mugs @ \$1.35 ea-24hr Service

Sponsored


Related searches: [white mug](#) [coffee mug](#) [mug root beer](#) [mug shot](#)




Representational
500 × 429 - 91k - jpg
[eagereyes.org](#)
[Find similar images](#)




Ceramic Happy Face
300 × 300 - 77k - jpg
[larose.com](#)
[Find similar images](#)




Here I go then, trying
600 × 600 - 35k - jpg
[beeper.wordpress.com](#)
[Find similar images](#)




The Chalk Mug »
304 × 314 - 17k - jpg
[coolest-gadgets.com](#)
[Find similar images](#)




mug
300 × 279 - 54k - jpg
[reynosawatch.org](#)




Bring your own
500 × 451 - 15k - jpg
[cookstownunited.ca](#)
[Find similar images](#)




ceramic mug
980 × 1024 - 30k - jpg
[diytrade.com](#)




Dual Purpose Drinking
490 × 428 - 16k - jpg
[freshome.com](#)
[Find similar images](#)




This coffee mug,
300 × 300 - 22k - jp
[gizmodo.com](#)
[Find similar images](#)




Back to Ceramic
400 × 400 - 8k - jpg
[freshpromotions.com.au](#)
[Find similar images](#)



Coffee Mug as a
303 × 301 - 10k - jpg
[dustbowl.wordpress.com](#)
[Find similar images](#)



SASS Life Member
300 × 302 - 6k - jpg
[sassnet.com](#)

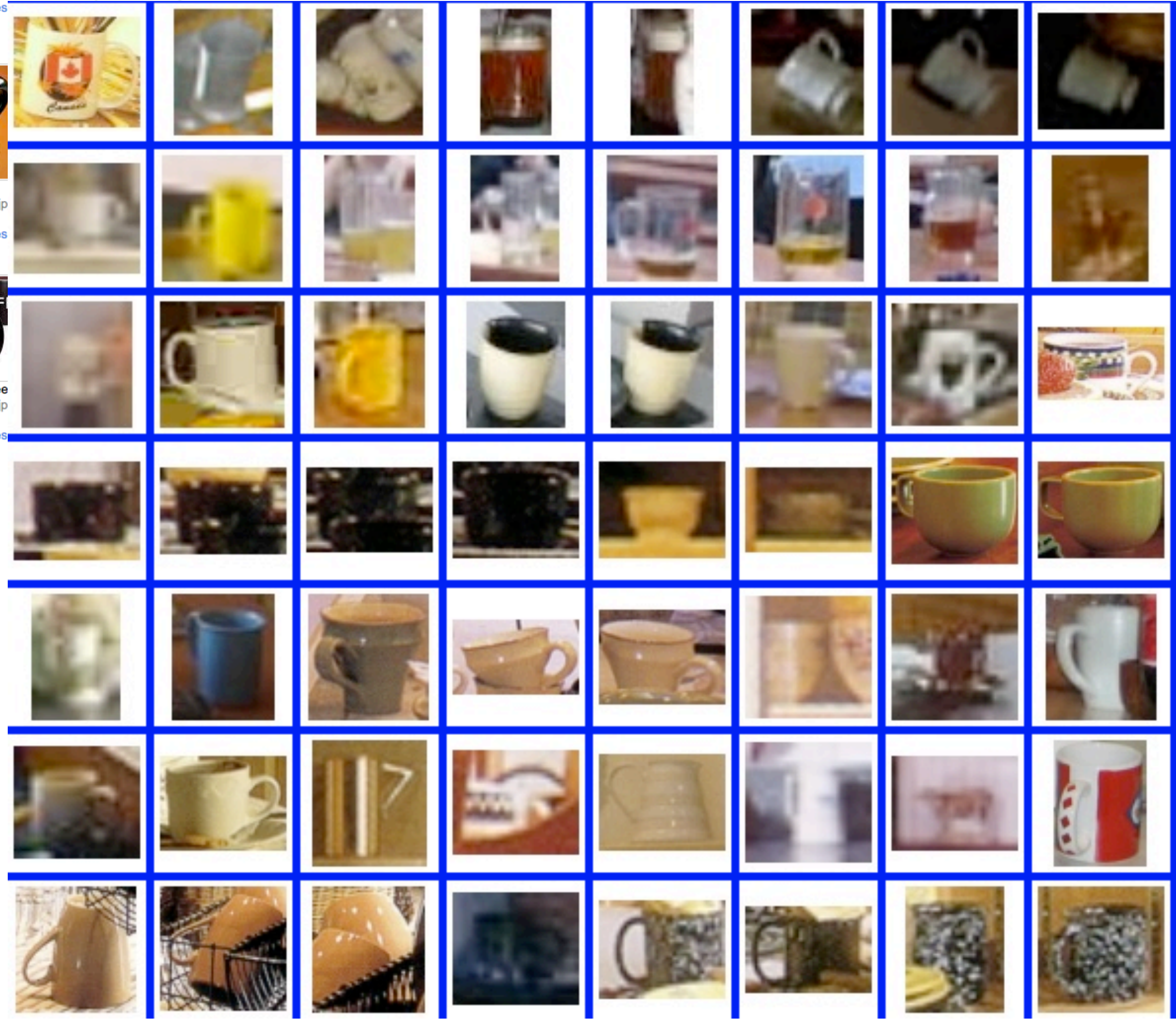


personalized coffee
400 × 343 - 15k - jp
[walyou.com](#)
[Find similar images](#)

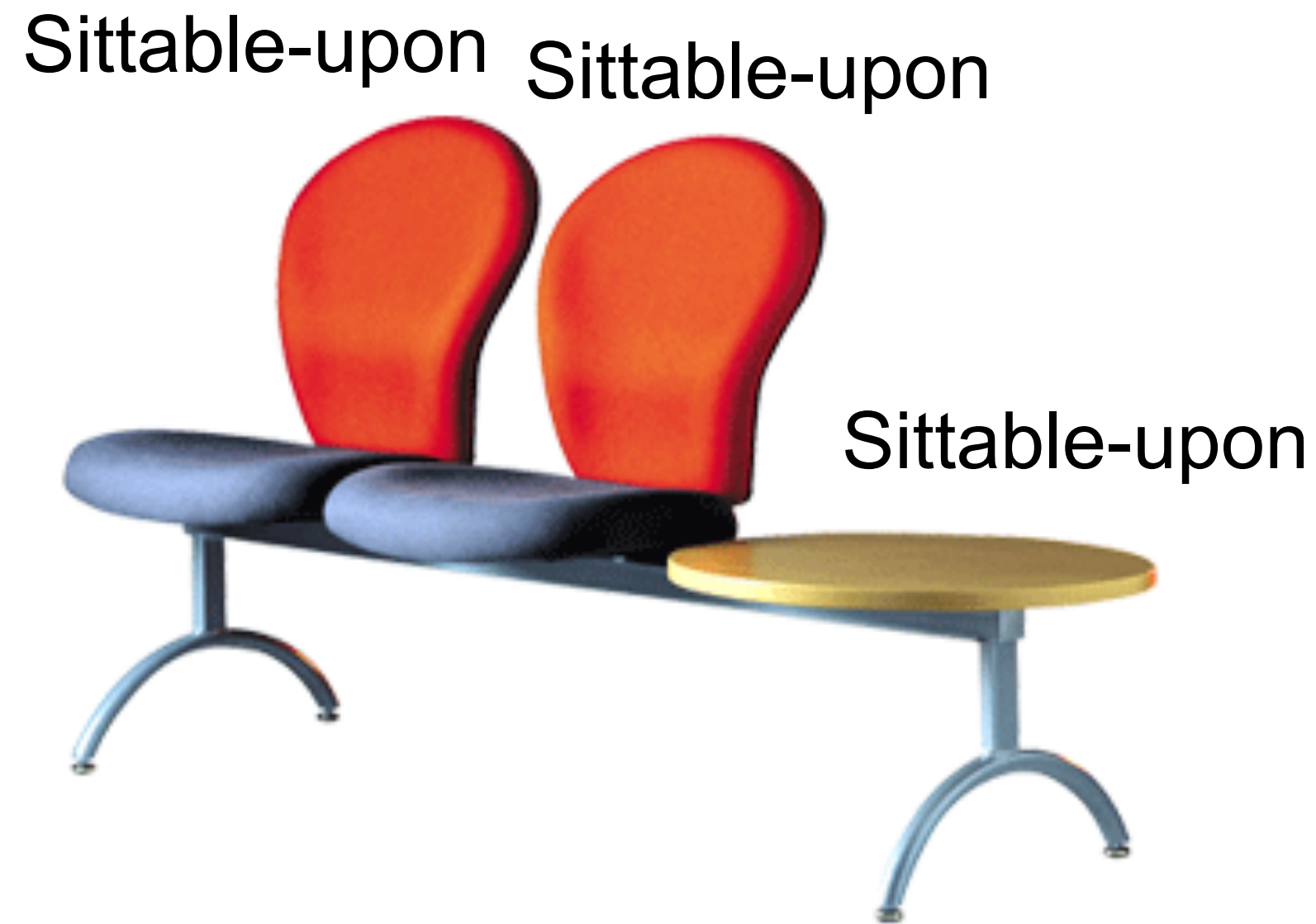
Google mugs

Dataset biases

Mugs from LabelMe



What is a chair?



It does not seem easy
to sit-upon this...



Some aspects of an object function can be perceived directly

- Functional form: Some forms clearly indicate to a function (“sittable-upon”, container, cutting device, ...)

Other questions

What if the object is present in the scene but invisible in the image?

What if there are infinite classes?

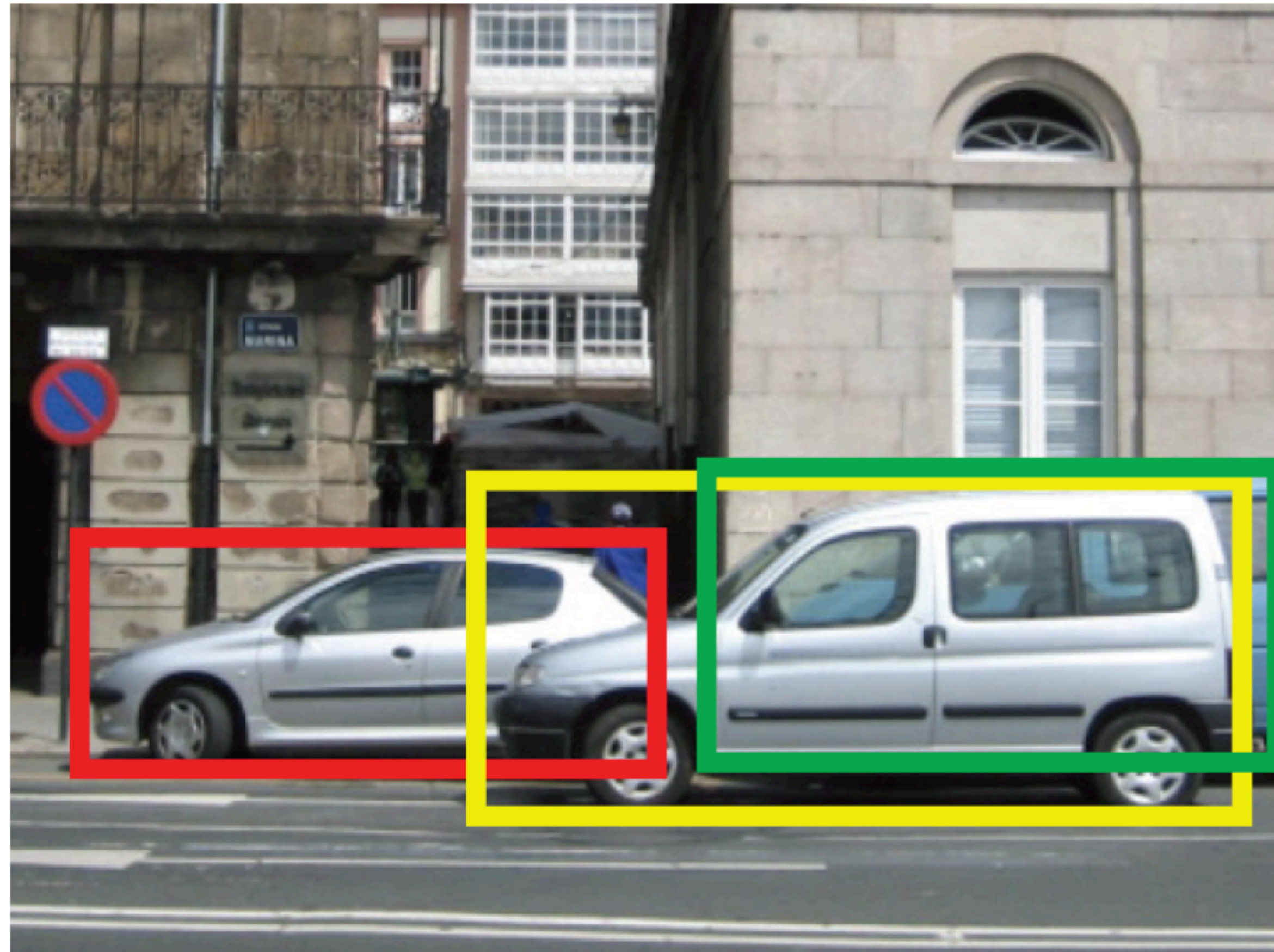
Is it possible to classify an image without localizing the object? How can we answer to the question “is object c present in the image?” without localizing the object?

Image classification performance could mislead us into believing that the classifier works, but it could be learning spurious correlations.

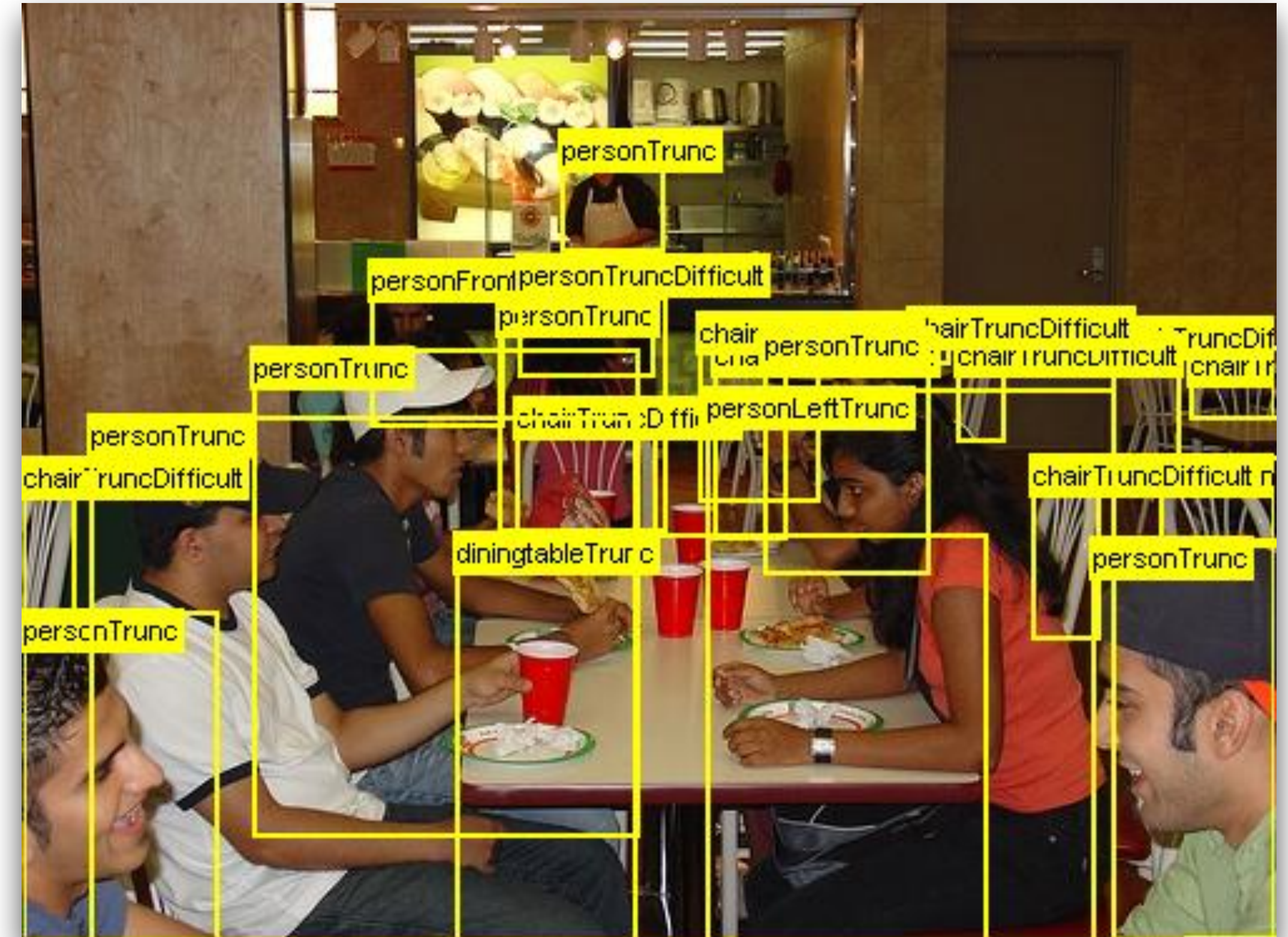
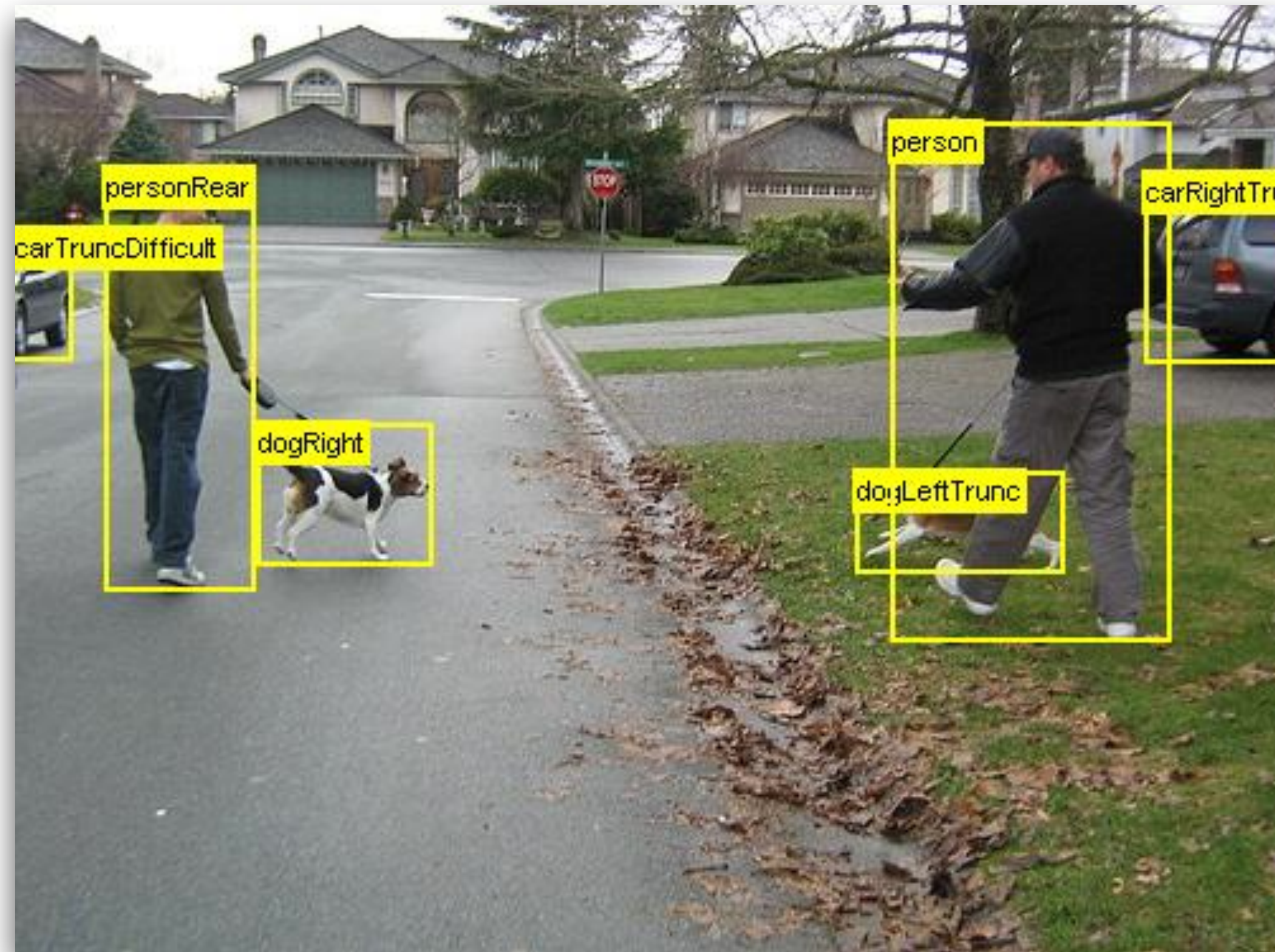
Object localization

Draw a box around each of the instances of class c in the input image.

Locate all cars in this image



PASCAL Visual Object Challenge



20 Object classes: aeroplane bike bird boat bottle bus car cat chair cow table dog horse motorbike person plant sheep sofa train tv

5000 training images

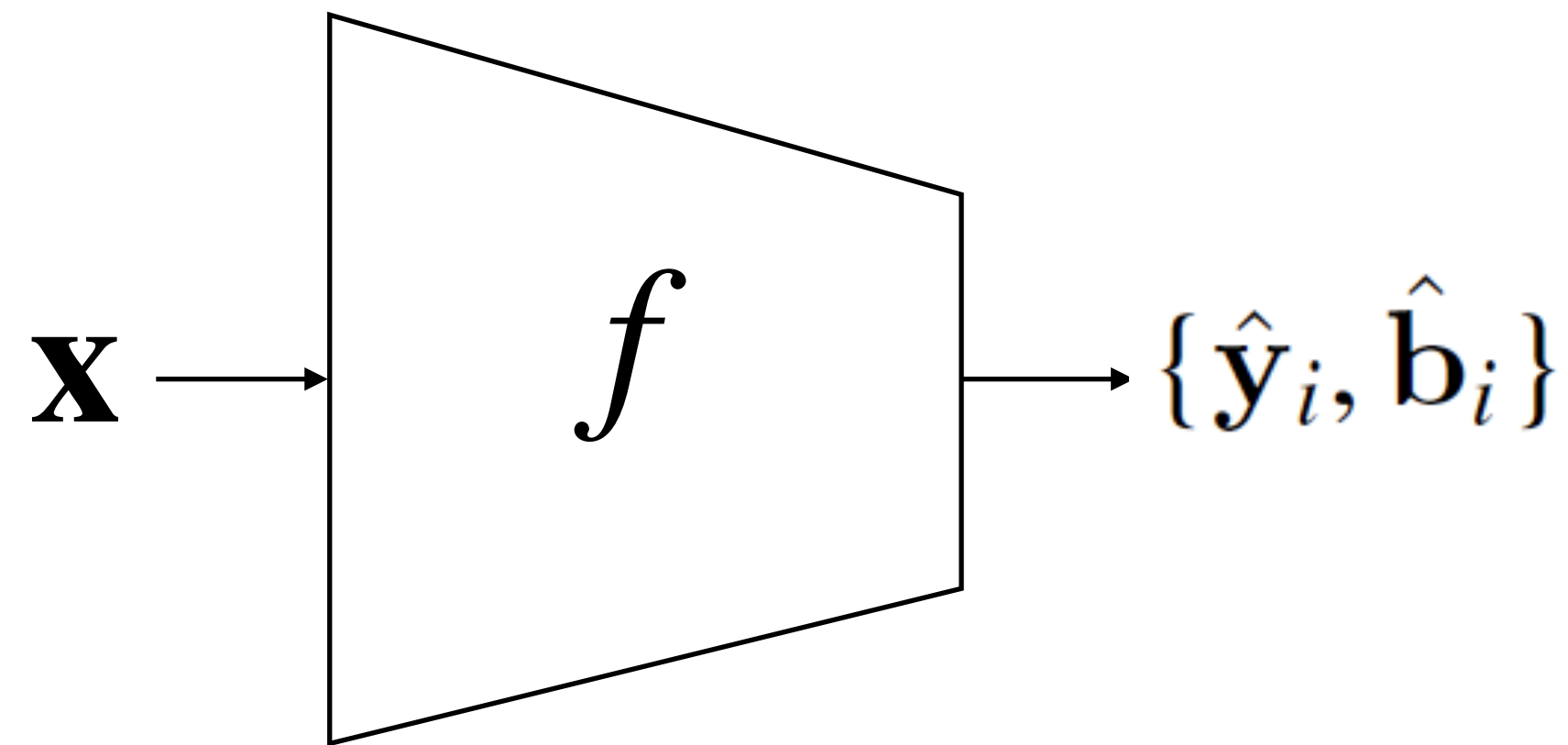
5000 testing images

Competition from 2005 - 2012

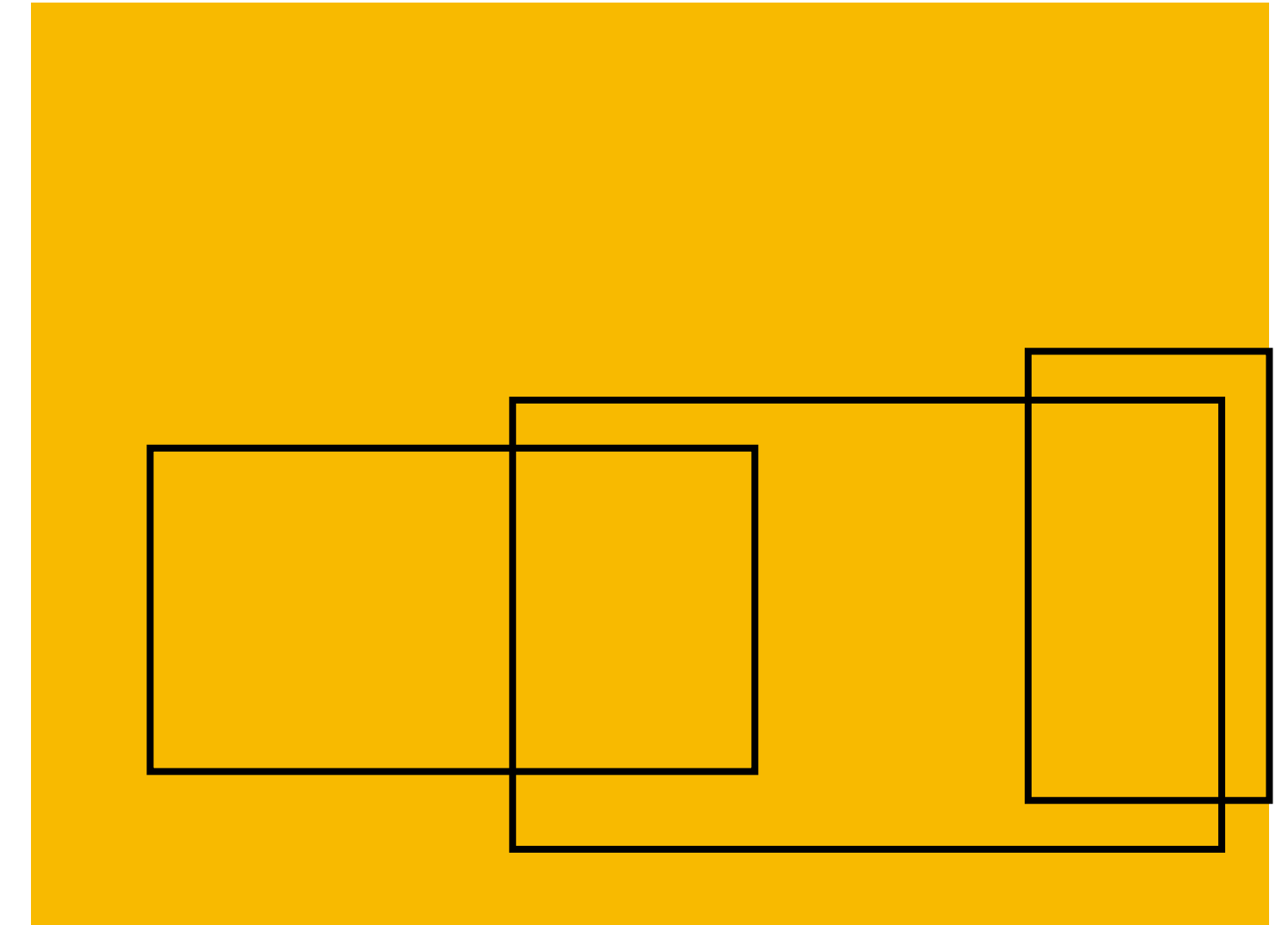
<http://host.robots.ox.ac.uk/pascal/VOC/>

Object localization: formulation

Our goal is a function f that outputs a set of bounding boxes, \mathbf{b} , and their classes \mathbf{y} :



$$\{\hat{\mathbf{y}}_i, \hat{\mathbf{b}}_i\} = f(\mathbf{x})$$



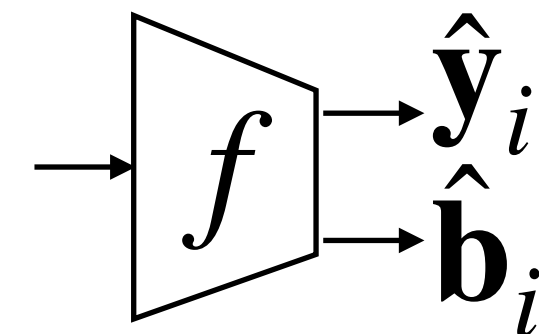
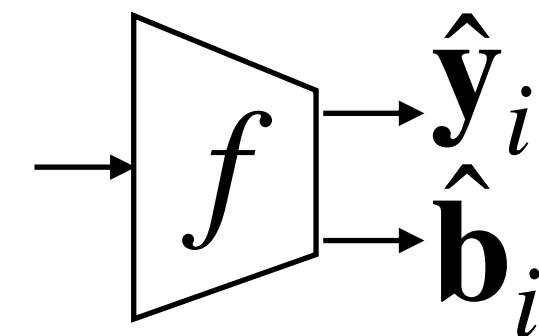
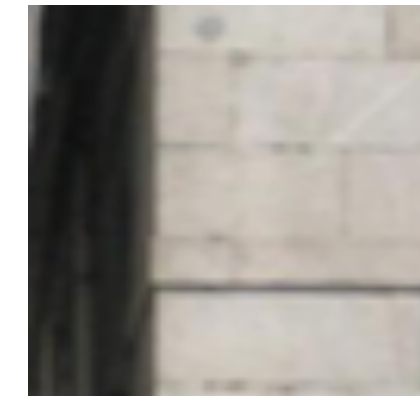
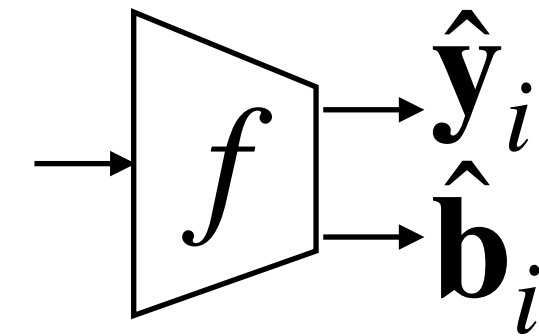
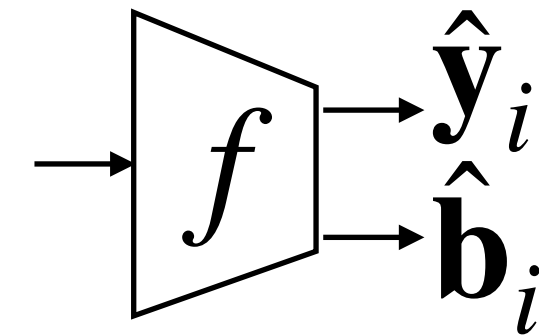
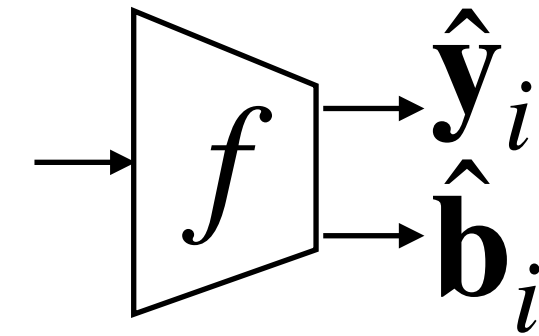
$$\mathbf{b} = [x_1, y_1, x_2, y_2]$$

Turning localization into classification

Input image



Region Proposals
(patches)



⋮



Turning localization into classification

Input image



Region Proposals
(patches)



Classify each patch

Turning localization into classification

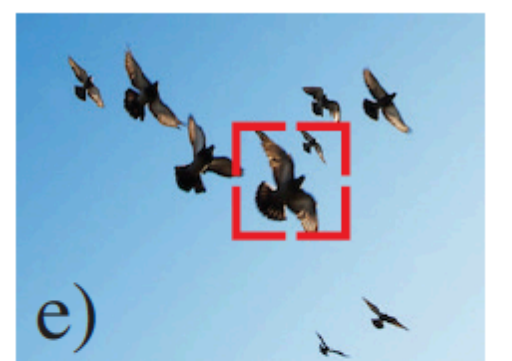
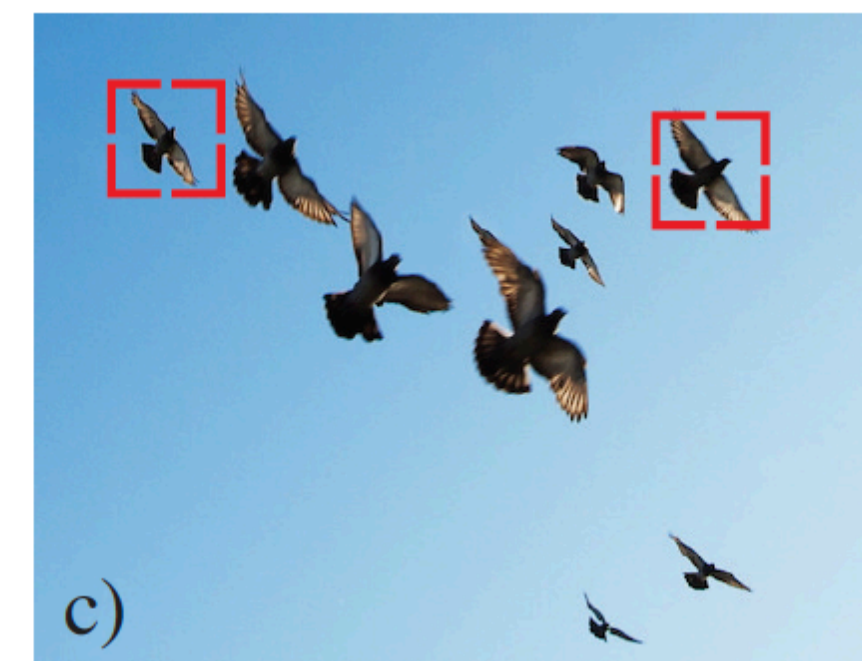
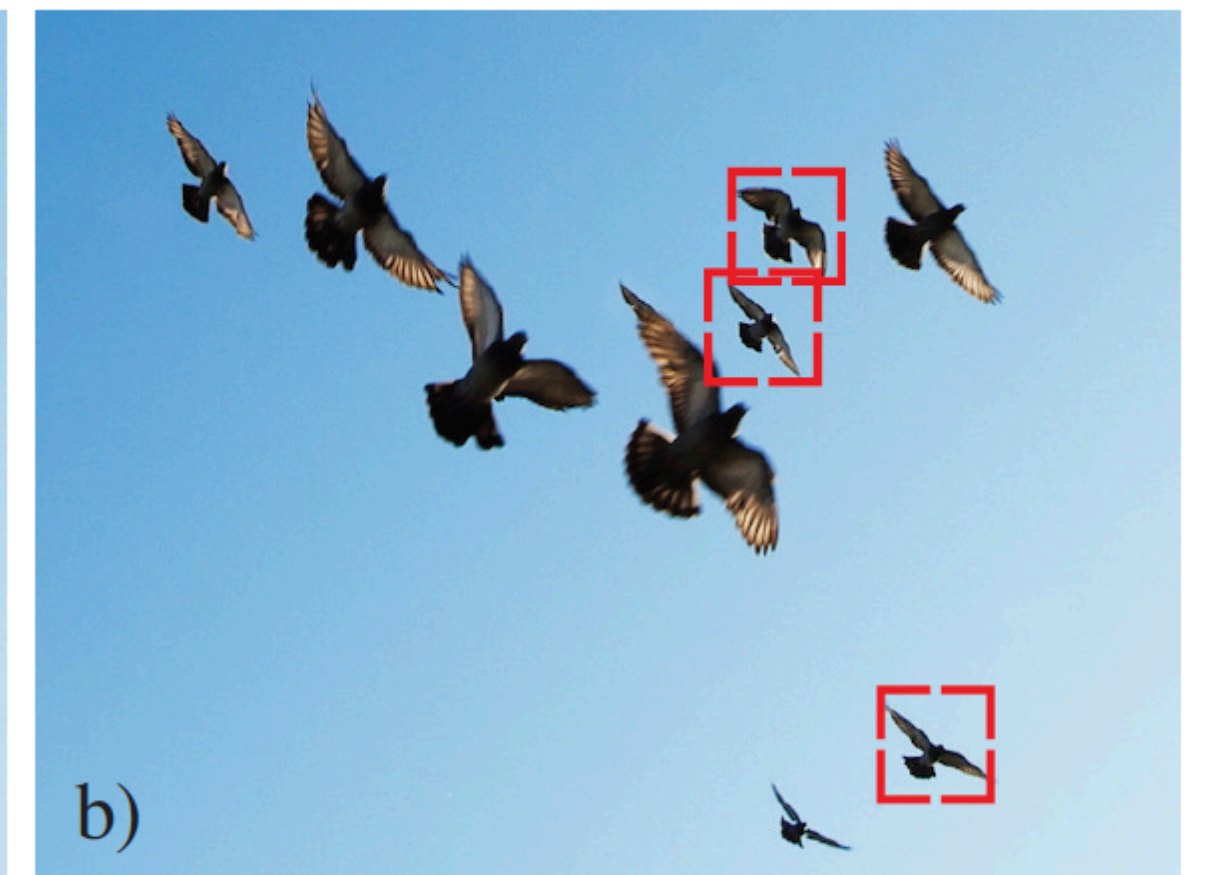
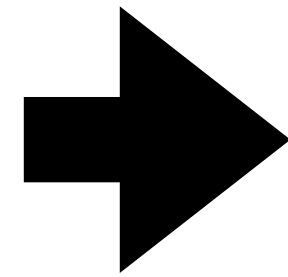


Image pyramids

Turning localization into classification

Window scanning approach



This can be computationally expensive

Turning localization into classification

Turning localization into classification

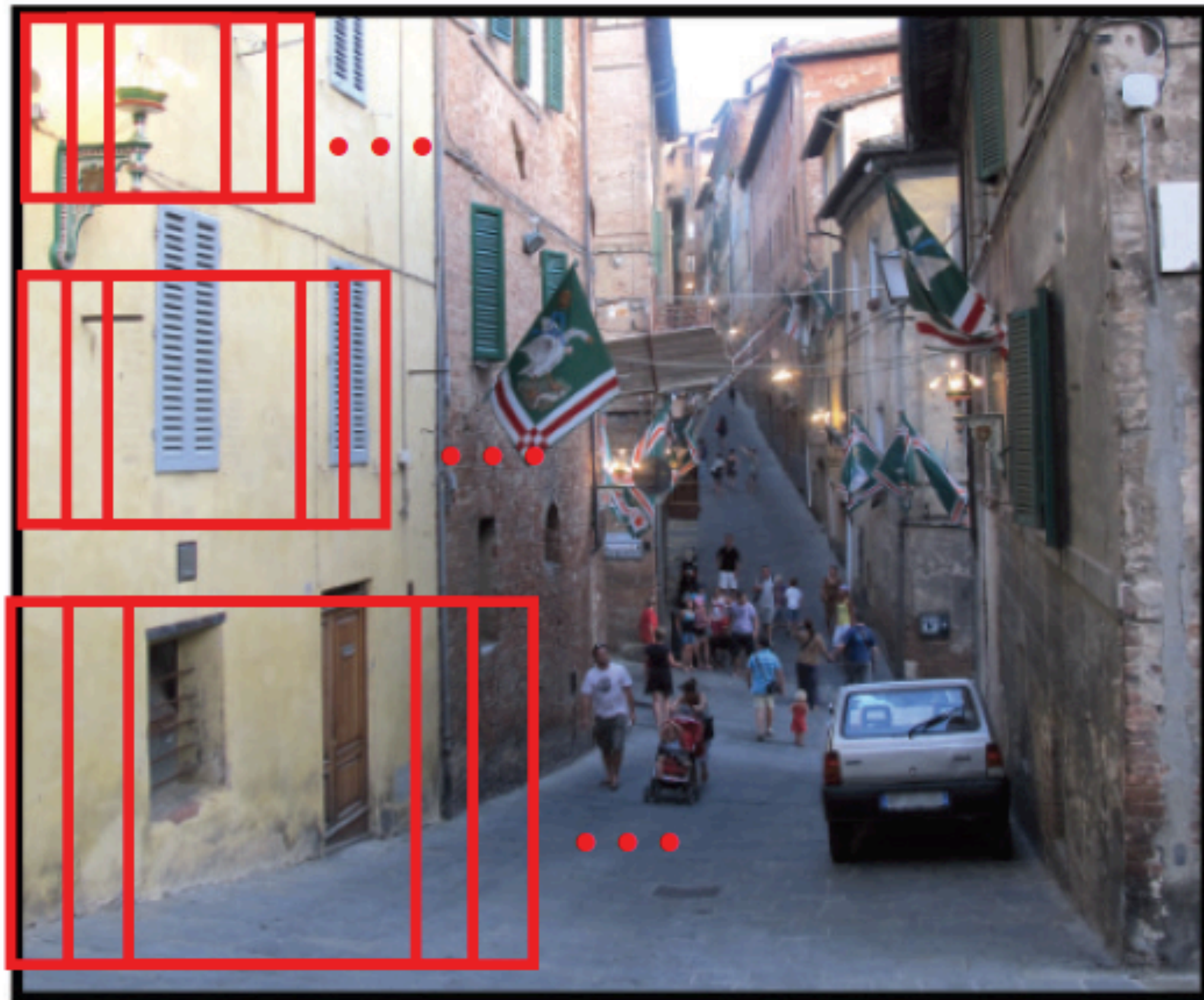
Selective search



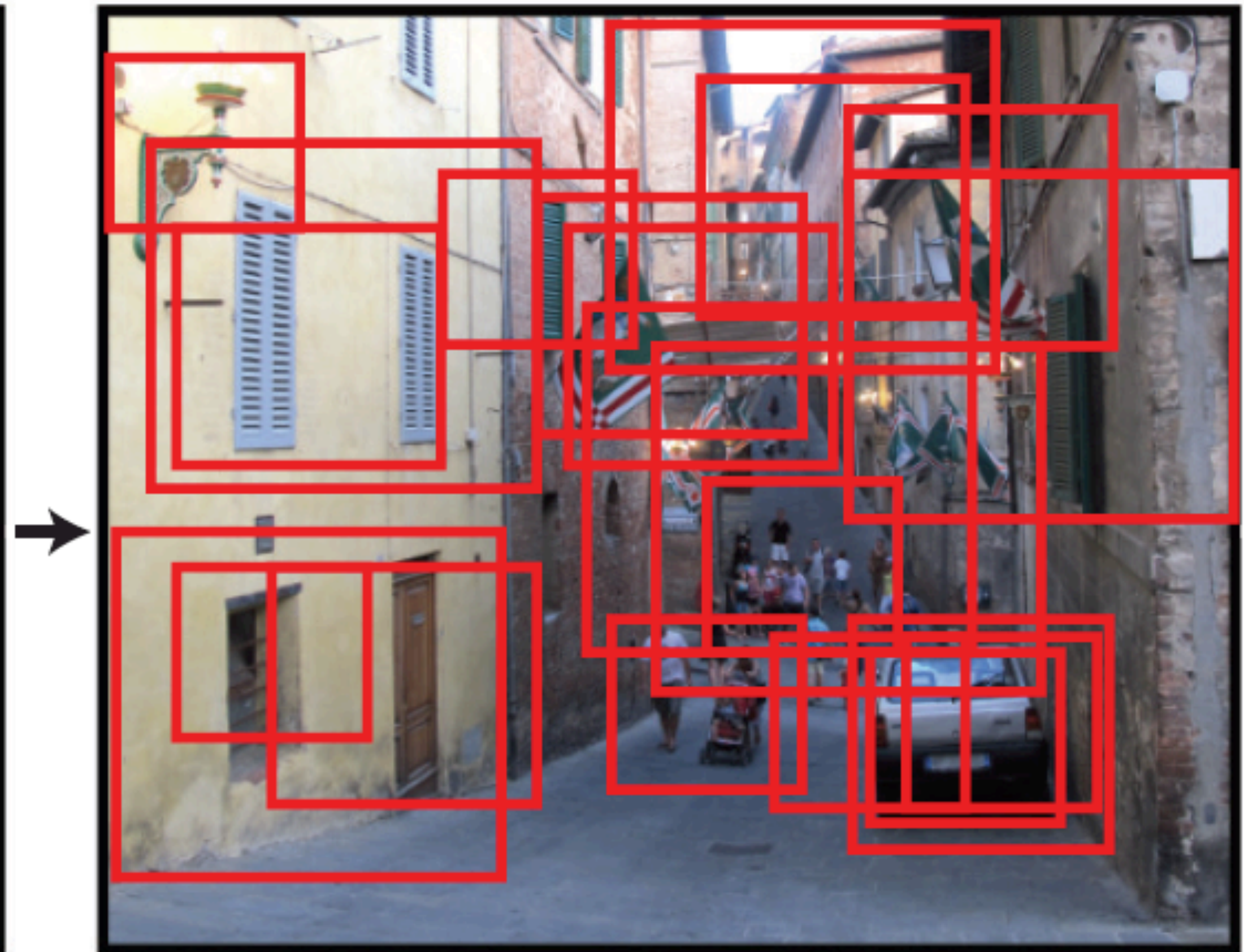
Selective search makes the process more efficient by proposing an initial set of bounding boxes that are good candidates to contain an object.

Turning localization into classification

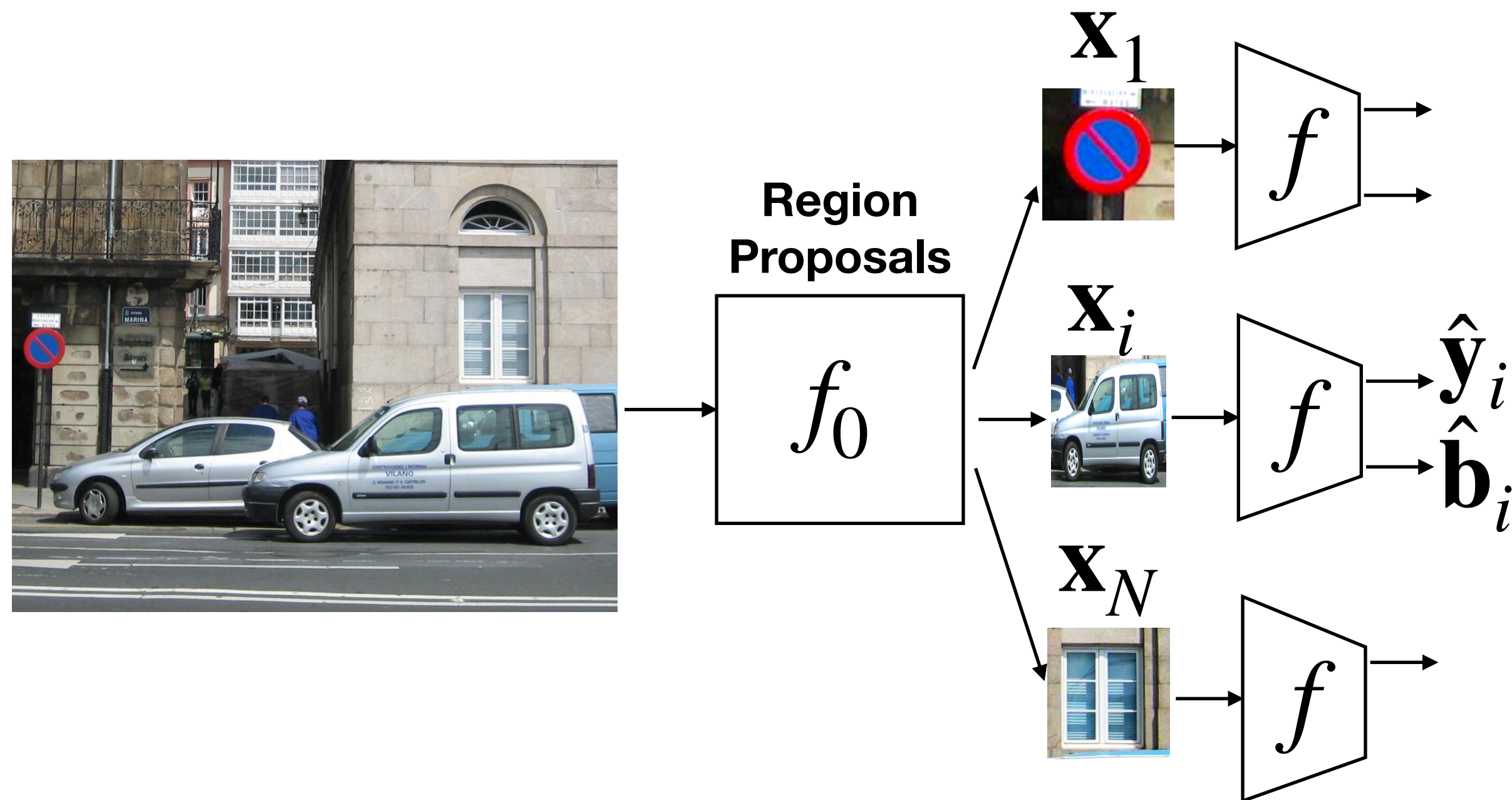
Window scanning



Bounding box proposals



Turning localization into classification



Loss function:

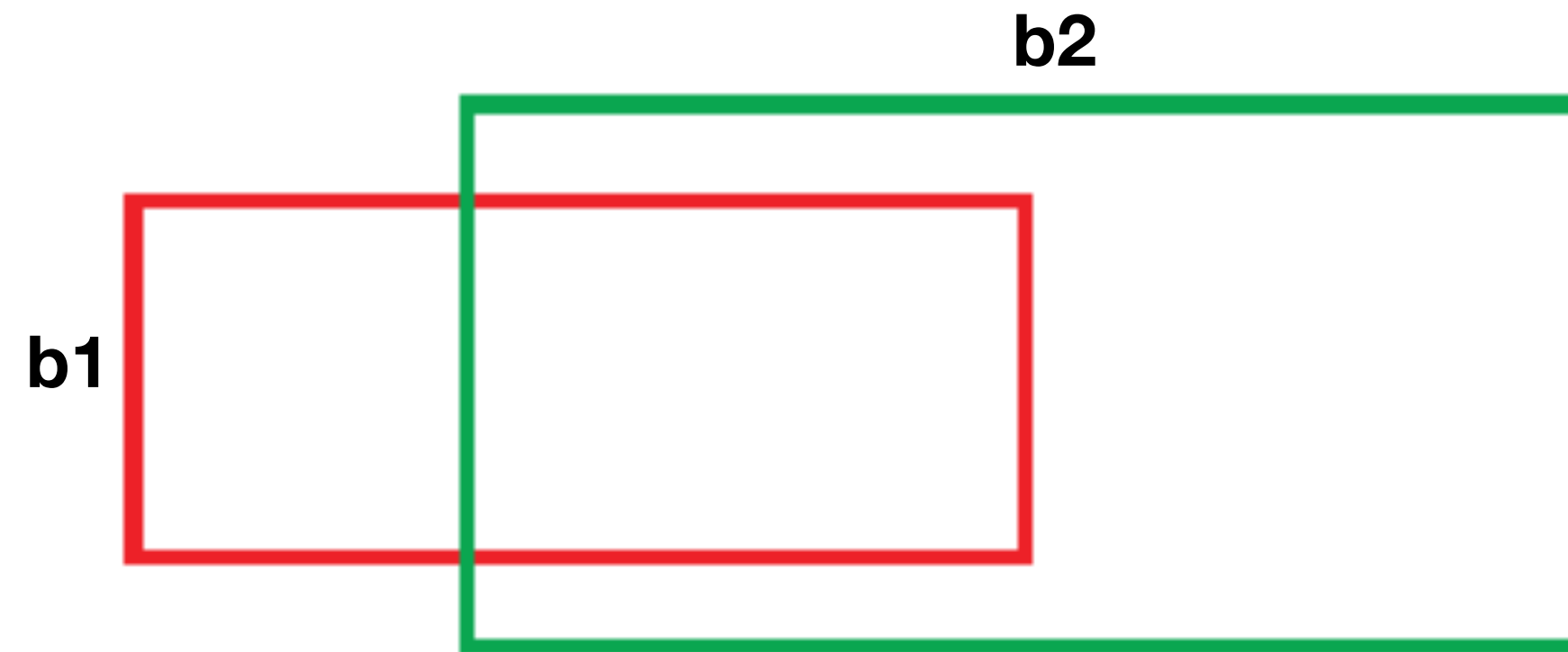
$$\mathcal{L}(\{\hat{\mathbf{b}}_i, \hat{y}_i\}, \{\mathbf{b}_i, y_i\}) = \mathcal{L}_{\text{cls}}(\hat{y}_i, y_i) + \lambda \mathbb{1}(y_i \neq 0) \mathcal{L}_{\text{loc}}(\hat{\mathbf{b}}_i, \mathbf{b}_i)$$

$$\mathcal{L}_{\text{cls}}(\hat{y}_i, y_i) = - \sum_{c=1}^K y_{c,i} \log(\hat{y}_{c,i})$$

We need a way to measure how different are two bounding boxes

Measuring similarity between bounding boxes

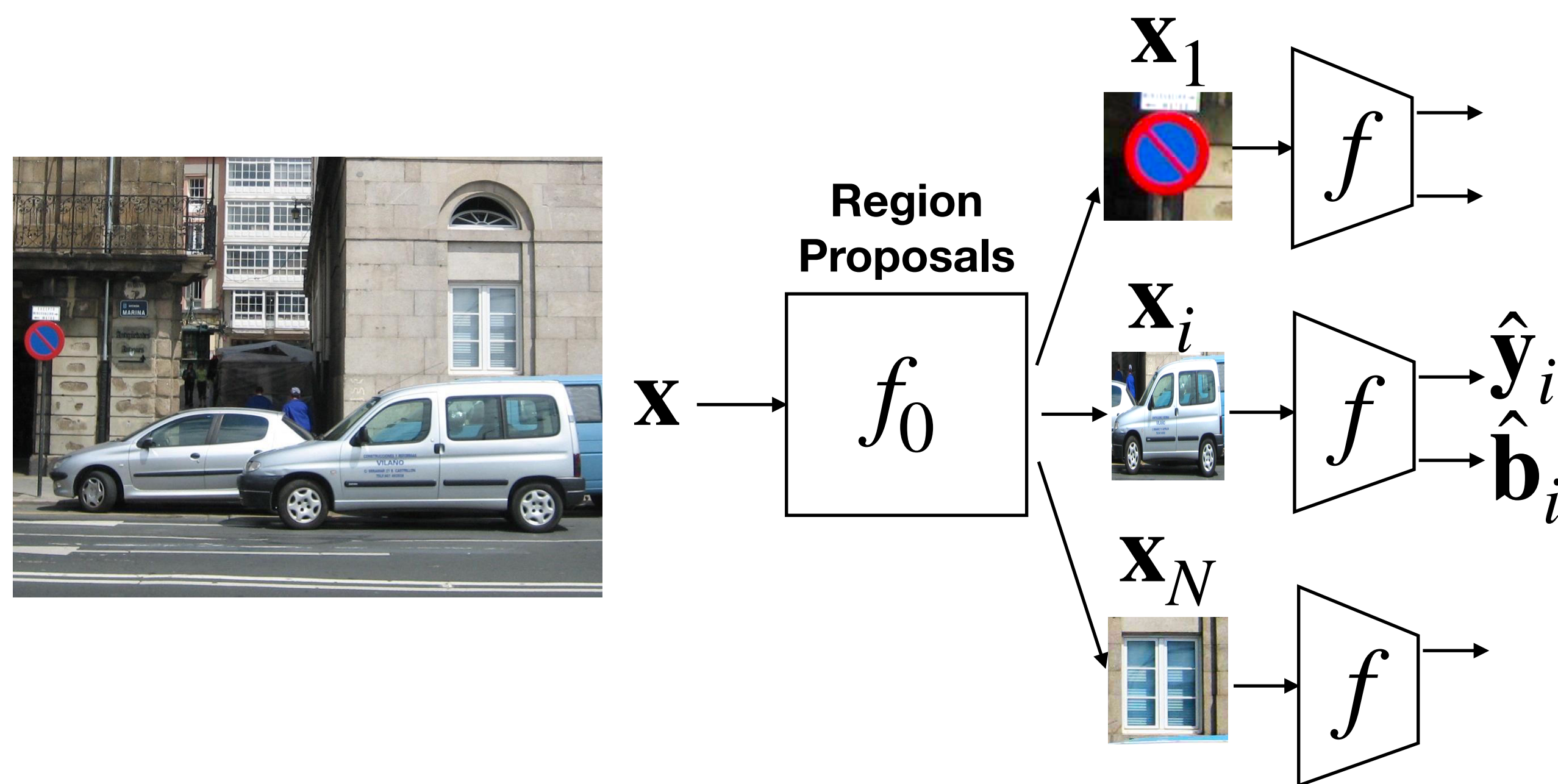
One typical measure of similarity between two bounding boxes is the **Intersection over Union** (IoU)



$$\text{IoU} = \frac{\text{Intersection}}{\text{Union}}$$



Turning localization into classification



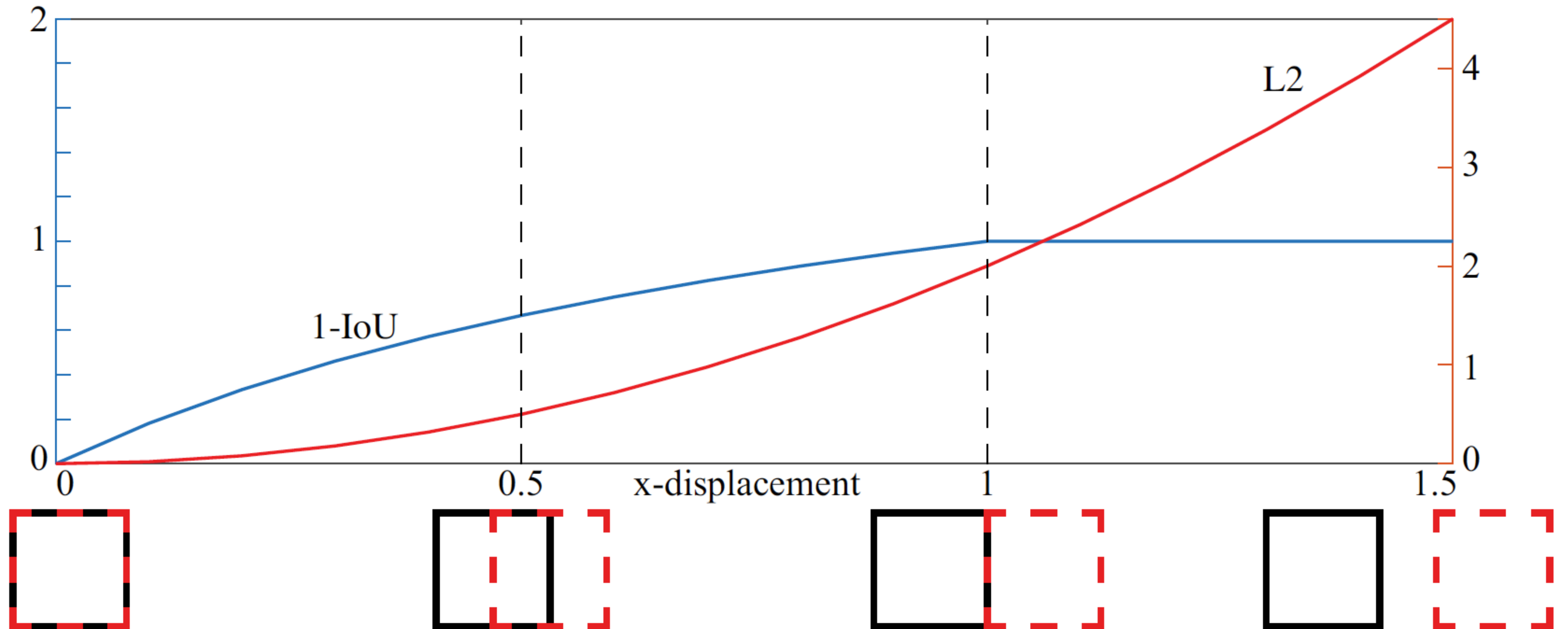
$$\mathcal{L}(\{\hat{\mathbf{b}}_i, \hat{y}_i\}, \{\mathbf{b}_i, y_i\}) = \mathcal{L}_{\text{cls}}(\hat{y}_i, y_i) + \lambda \mathbb{1}(y_i \neq 0) \mathcal{L}_{\text{loc}}(\hat{\mathbf{b}}_i, \mathbf{b}_i)$$

$$\mathcal{L}_{\text{cls}}(\hat{y}_i, y_i) = - \sum_{c=1}^K y_{c,i} \log(\hat{y}_{c,i})$$

$$\mathcal{L}_{\text{loc}}(\hat{\mathbf{b}}, \mathbf{b}) = 1 - \text{IoU}(\hat{\mathbf{b}}, \mathbf{b})$$

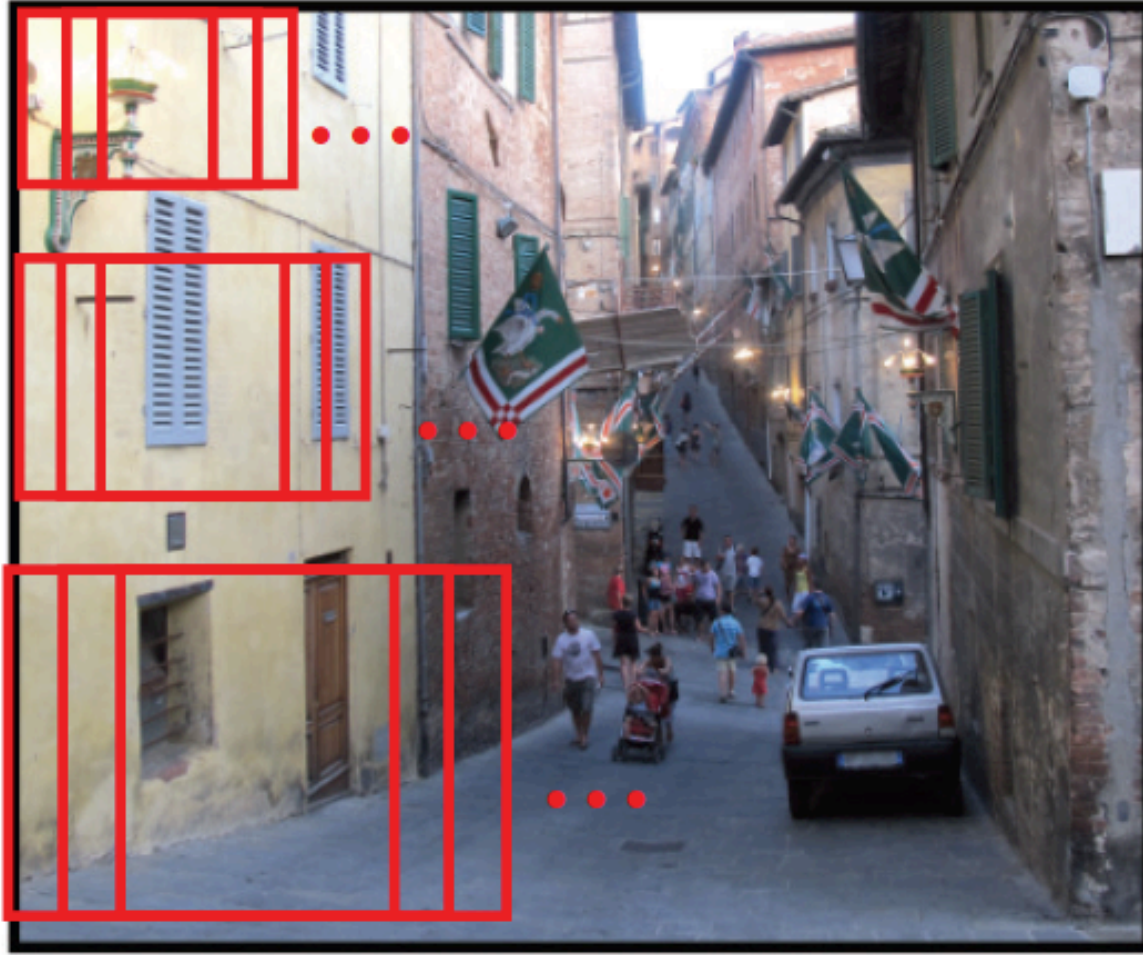
$$\mathcal{L}_{\text{loc}}(\hat{\mathbf{b}}, \mathbf{b}) = (\hat{x}_1 - x_1)^2 + (\hat{x}_2 - x_2)^2 + (\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2$$

Measuring similarity between bounding boxes

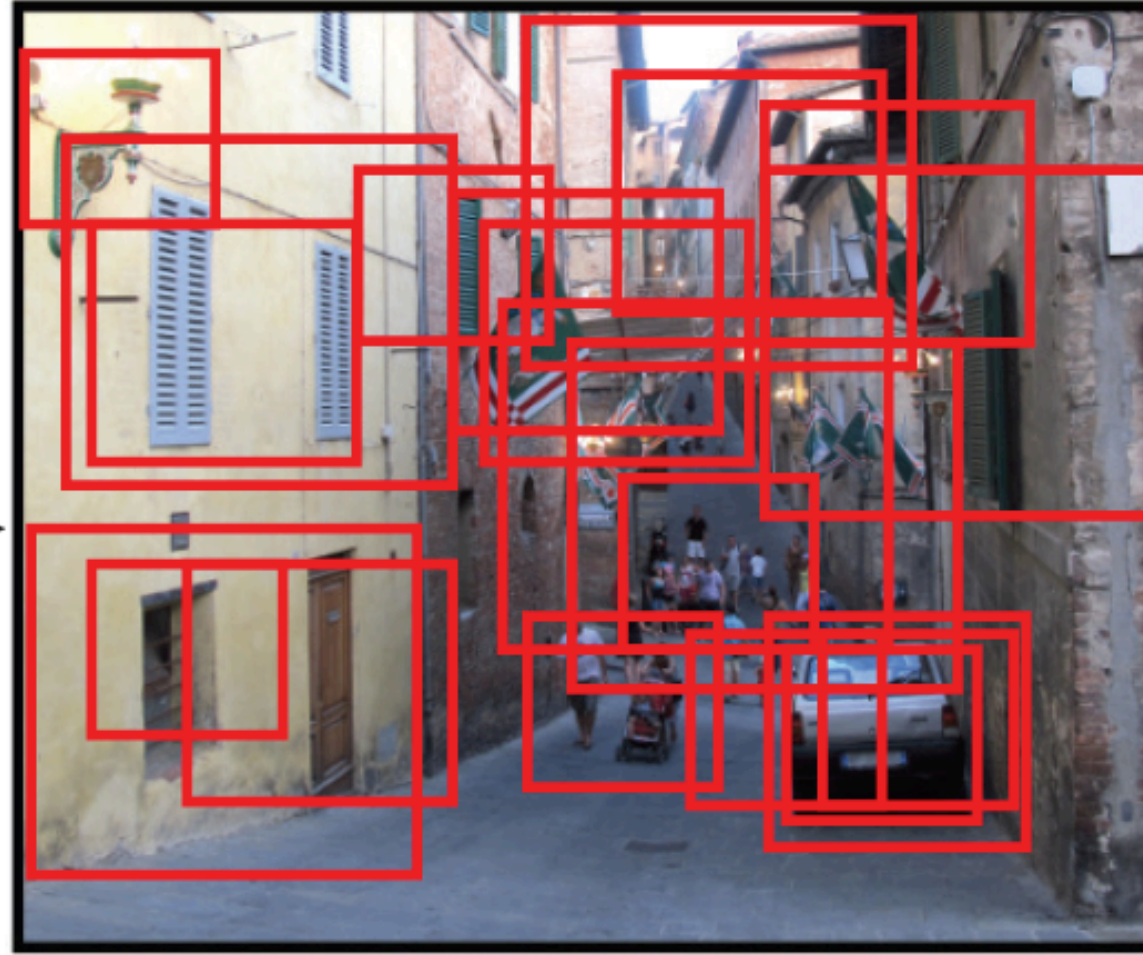


Non-maximal suppression

Window scanning



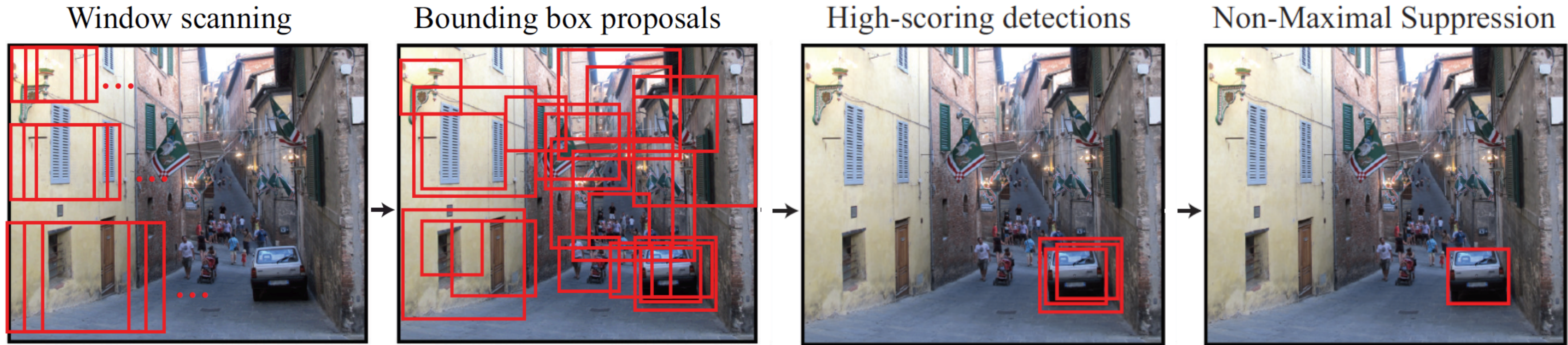
Bounding box proposals



High-scoring detections



Non-maximal suppression



1. Take the highest confidence bounding box from the set S and add it to the final set S^*
2. Remove from S the selected bounding box and all the bounding boxes with an IoU larger than a threshold.
3. go to step 1 until S is empty.

R-CNN, Fast R-CNN, Faster R-CNN

Rich feature hierarchies for accurate object detection and semantic segmentation

Tech report (v5)

Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik
UC Berkeley

{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 compare R-CNN to OverFeat, a recently proposed sliding-window detector based on a similar CNN architecture. We find that R-CNN outperforms OverFeat by a large margin on the 200-class ILSVRC2013 detection dataset. 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 [29] and HOG [7]. But if we look at performance on the canonical visual recognition task, PASCAL VOC object detection [15], 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 hier-

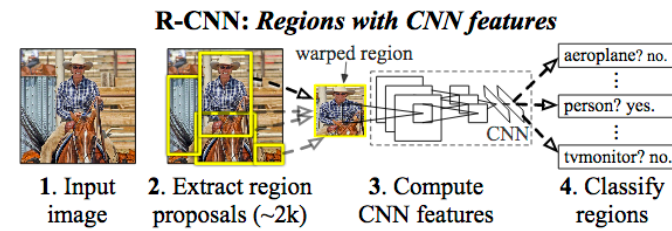


Figure 1: Object detection system overview. Our system (1) takes an input image, (2) extracts around 2000 bottom-up region proposals, (3) computes features for each proposal using a large convolutional neural network (CNN), and then (4) classifies each region using class-specific linear SVMs. R-CNN achieves a mean average precision (mAP) of **53.7% on PASCAL VOC 2010**. For comparison, [39] 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%. On the 200-class ILSVRC2013 detection dataset, R-CNN's mAP is **31.4%**, a large improvement over OverFeat [34], which had the previous best result at 24.3%.

archical, multi-stage processes for computing features that are even more informative for visual recognition.

Fukushima's “neocognitron” [19], 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. Building on Rumelhart et al. [33], LeCun et al. [26] showed that stochastic gradient descent via back-propagation was effective for training convolutional neural networks (CNNs), a class of models that extend the neocognitron.

CNNs saw heavy use in the 1990s (e.g., [27]), but then fell out of fashion with the rise of support vector machines. In 2012, Krizhevsky et al. [25] 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

Fast R-CNN

Ross Girshick
Microsoft Research
rbg@microsoft.com

Abstract

This paper proposes a Fast Region-based Convolutional Network method (Fast R-CNN) for object detection. Fast R-CNN builds on previous work to efficiently classify object proposals using deep convolutional networks. Compared to previous work, Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection accuracy. Fast R-CNN trains the very deep VGG16 network $9\times$ faster than R-CNN, is $213\times$ faster at test-time, and achieves a higher mAP on PASCAL VOC 2012. Compared to SPPnet, Fast R-CNN trains VGG16 $3\times$ faster, tests $10\times$ faster, and is more accurate. Fast R-CNN is implemented in Python and C++ (using Caffe) and is available under the open-source MIT License at <https://github.com/rbgirshick/fast-rcnn>.

1. Introduction

Recently, deep ConvNets [14, 16] have significantly improved image classification [14] and object detection [9, 19] accuracy. Compared to image classification, object detection is a more challenging task that requires more complex methods to solve. Due to this complexity, current approaches (e.g., [9, 11, 19, 25]) train models in multi-stage pipelines that are slow and inelegant.

Complexity arises because detection requires the accurate localization of objects, creating two primary challenges. First, numerous candidate object locations (often called “proposals”) must be processed. Second, these candidates provide only rough localization that must be refined to achieve precise localization. Solutions to these problems often compromise speed, accuracy, or simplicity.

In this paper, we streamline the training process for state-of-the-art ConvNet-based object detectors [9, 11]. We propose a single-stage training algorithm that jointly learns to classify object proposals and refine their spatial locations.

The resulting method can train a very deep detection network (VGG16 [20]) $9\times$ faster than R-CNN [9] and $3\times$ faster than SPPnet [11]. At runtime, the detection network processes images in 0.3s (excluding object proposal time)

while achieving top accuracy on PASCAL VOC 2012 [7] with a mAP of 66% (vs. 62% for R-CNN).¹

1.1. R-CNN and SPPnet

The Region-based Convolutional Network method (R-CNN) [9] achieves excellent object detection accuracy by using a deep ConvNet to classify object proposals. R-CNN, however, has notable drawbacks:

- 1. Training is a multi-stage pipeline.** R-CNN first fine-tunes a ConvNet on object proposals using log loss. Then, it fits SVMs to ConvNet features. These SVMs act as object detectors, replacing the softmax classifier learnt by fine-tuning. In the third training stage, bounding-box regressors are learned.
- 2. Training is expensive in space and time.** For SVM and bounding-box regressor training, features are extracted from each object proposal in each image and written to disk. With very deep networks, such as VGG16, this process takes 2.5 GPU-days for the 5k images of the VOC07 trainval set. These features require hundreds of gigabytes of storage.
- 3. Object detection is slow.** At test-time, features are extracted from each object proposal in each test image. Detection with VGG16 takes 47s / image (on a GPU).

R-CNN is slow because it performs a ConvNet forward pass for each object proposal, without sharing computation. Spatial pyramid pooling networks (SPPnets) [11] were proposed to speed up R-CNN by sharing computation. The SPPnet method computes a convolutional feature map for the entire input image and then classifies each object proposal using a feature vector extracted from the shared feature map. Features are extracted for a proposal by max-pooling the portion of the feature map inside the proposal into a fixed-size output (e.g., 6×6). Multiple output sizes are pooled and then concatenated as in spatial pyramid pooling [15]. SPPnet accelerates R-CNN by 10 to $100\times$ at test time. Training time is also reduced by $3\times$ due to faster proposal feature extraction.

¹All timings use one Nvidia K40 GPU overclocked to 875 MHz.

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

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

Abstract—State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet [1] and Fast R-CNN [2] have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck. In this work, we introduce a *Region Proposal Network* (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. We further merge RPN and Fast R-CNN into a single network by sharing their convolutional features—using the recently popular terminology of neural networks with “attention” mechanisms, the RPN component tells the unified network where to look. For the very deep VGG-16 model [3], our detection system has a frame rate of 5fps (including all steps) on a GPU, while achieving state-of-the-art object detection accuracy on PASCAL VOC 2007, 2012, and MS COCO datasets with only 300 proposals per image. In ILSVRC and COCO 2015 competitions, Faster R-CNN and RPN are the foundations of the 1st-place winning entries in several tracks. Code has been made publicly available.

Index Terms—Object Detection, Region Proposal, Convolutional Neural Network.

1 INTRODUCTION

Recent advances in object detection are driven by the success of region proposal methods (e.g., [4]) and region-based convolutional neural networks (R-CNNs) [5]. Although region-based CNNs were computationally expensive as originally developed in [5], their cost has been drastically reduced thanks to sharing convolutions across proposals [1], [2]. The latest incarnation, Fast R-CNN [2], achieves near real-time rates using very deep networks [3], *when ignoring the time spent on region proposals*. Now, proposals are the test-time computational bottleneck in state-of-the-art detection systems.

Region proposal methods typically rely on inexpensive features and economical inference schemes. Selective Search [4], one of the most popular methods, greedily merges superpixels based on engineered low-level features. Yet when compared to efficient detection networks [2], Selective Search is an order of magnitude slower, at 2 seconds per image in a CPU implementation. EdgeBoxes [6] currently provides the best tradeoff between proposal quality and speed, at 0.2 seconds per image. Nevertheless, the region proposal step still consumes as much running time as the detection network.

• S. Ren is with University of Science and Technology of China, Hefei, China. This work was done when S. Ren was an intern at Microsoft Research. Email: saren@mail.ustc.edu.cn
• K. He and J. Sun are with Visual Computing Group, Microsoft Research. E-mail: {kahe,jiansun}@microsoft.com
• R. Girshick is with Facebook AI Research. The majority of this work was done when R. Girshick was with Microsoft Research. E-mail: rbg@fb.com

One may note that fast region-based CNNs take advantage of GPUs, while the region proposal methods used in research are implemented on the CPU, making such runtime comparisons inequitable. An obvious way to accelerate proposal computation is to re-implement it for the GPU. This may be an effective engineering solution, but re-implementation ignores the down-stream detection network and therefore misses important opportunities for sharing computation.

In this paper, we show that an algorithmic change—computing proposals with a deep convolutional neural network—leads to an elegant and effective solution where proposal computation is nearly cost-free given the detection network's computation. To this end, we introduce novel *Region Proposal Networks* (RPNs) that share convolutional layers with state-of-the-art object detection networks [1], [2]. By sharing convolutions at test-time, the marginal cost for computing proposals is small (e.g., 10ms per image).

Our observation is that the convolutional feature maps used by region-based detectors, like Fast R-CNN, can also be used for generating region proposals. On top of these convolutional features, we construct an RPN by adding a few additional convolutional layers that simultaneously regress region bounds and objectness scores at each location on a regular grid. The RPN is thus a kind of fully convolutional network (FCN) [7] and can be trained end-to-end specifically for the task of generating detection proposals.

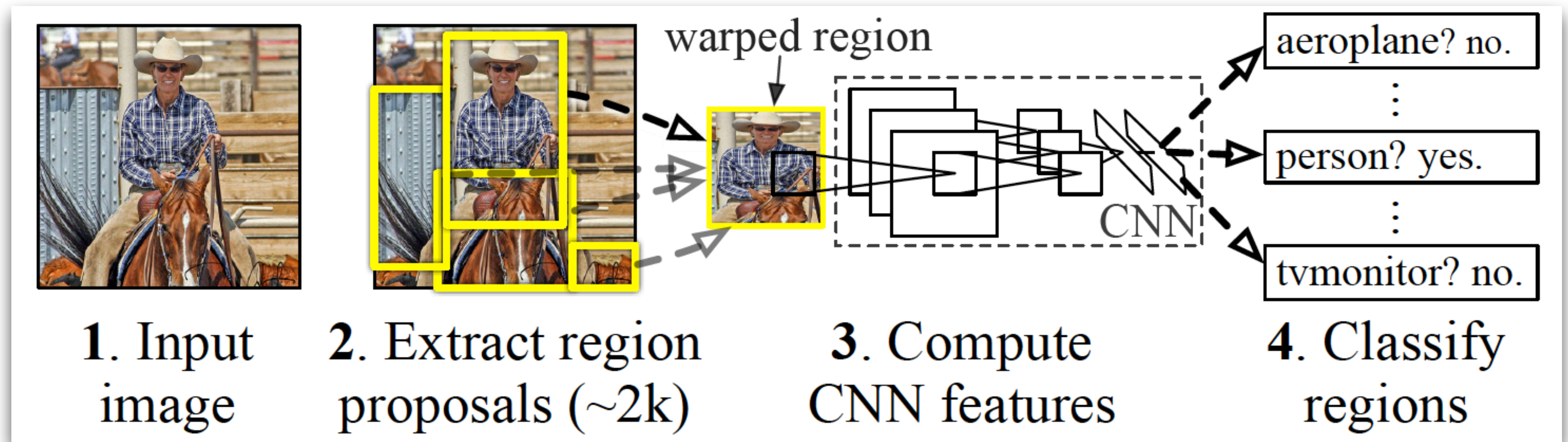
RPNs are designed to efficiently predict region proposals with a wide range of scales and aspect ratios. In contrast to prevalent methods [8], [9], [1], [2] that use

<https://arxiv.org/pdf/1311.2524.pdf>

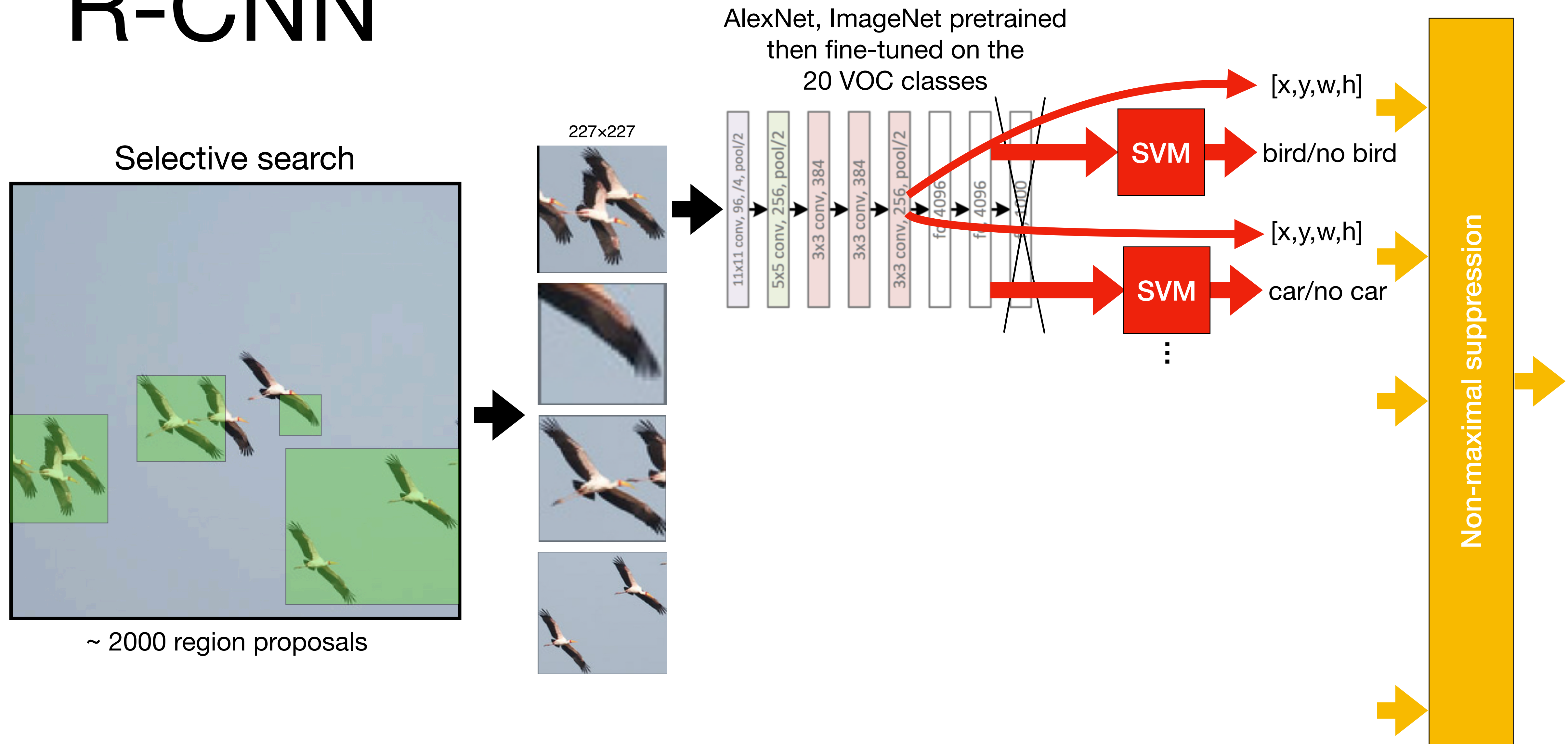
<https://arxiv.org/pdf/1504.08083.pdf>

<https://arxiv.org/pdf/1506.01497.pdf>

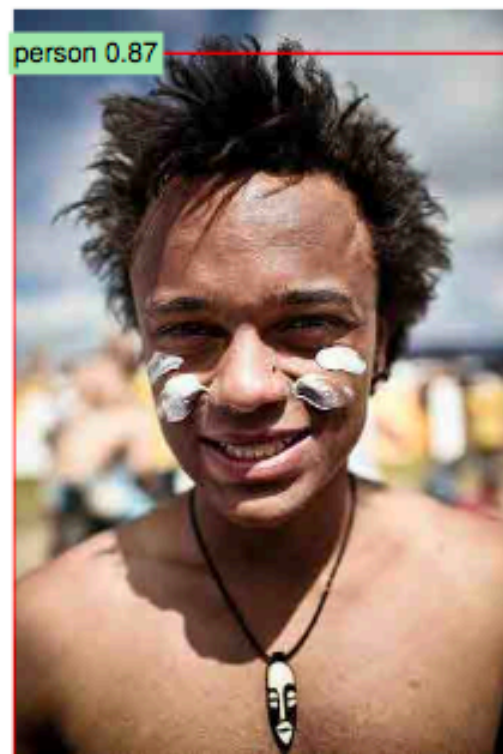
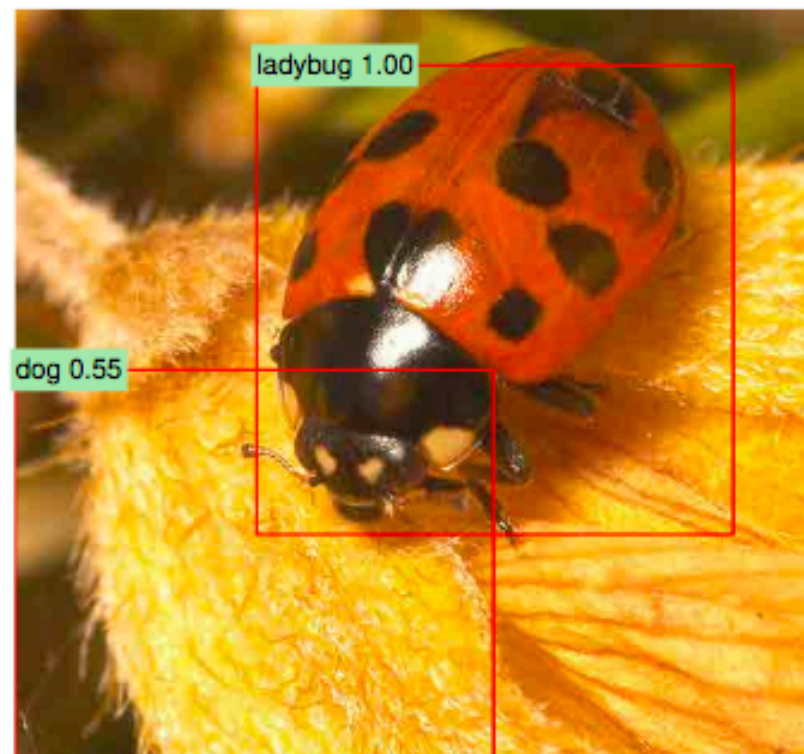
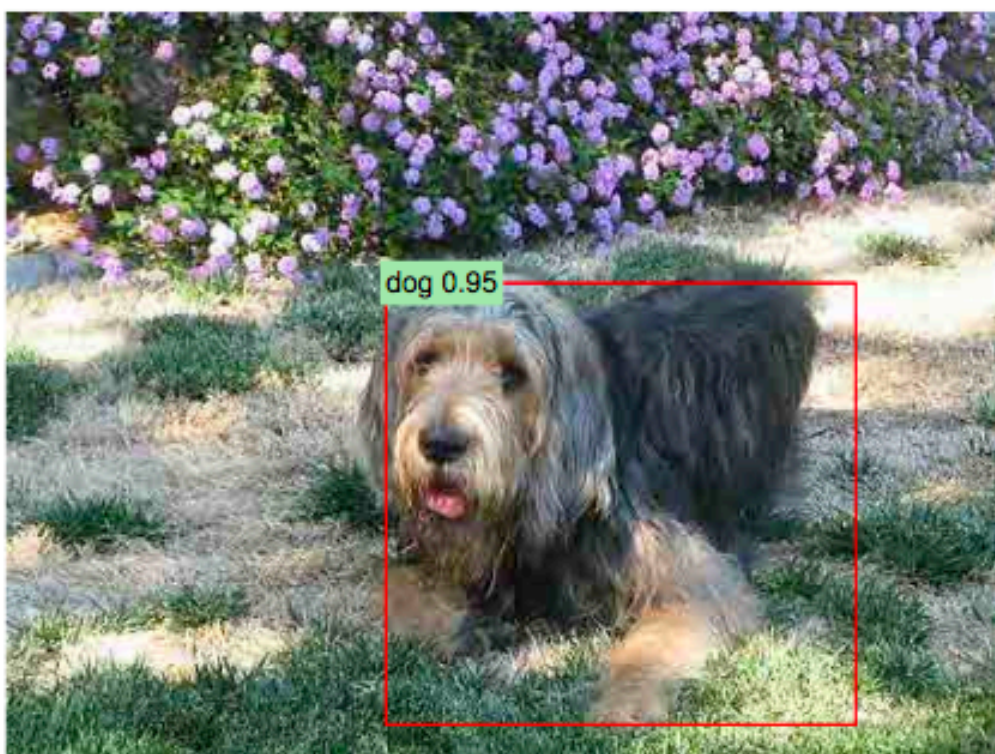
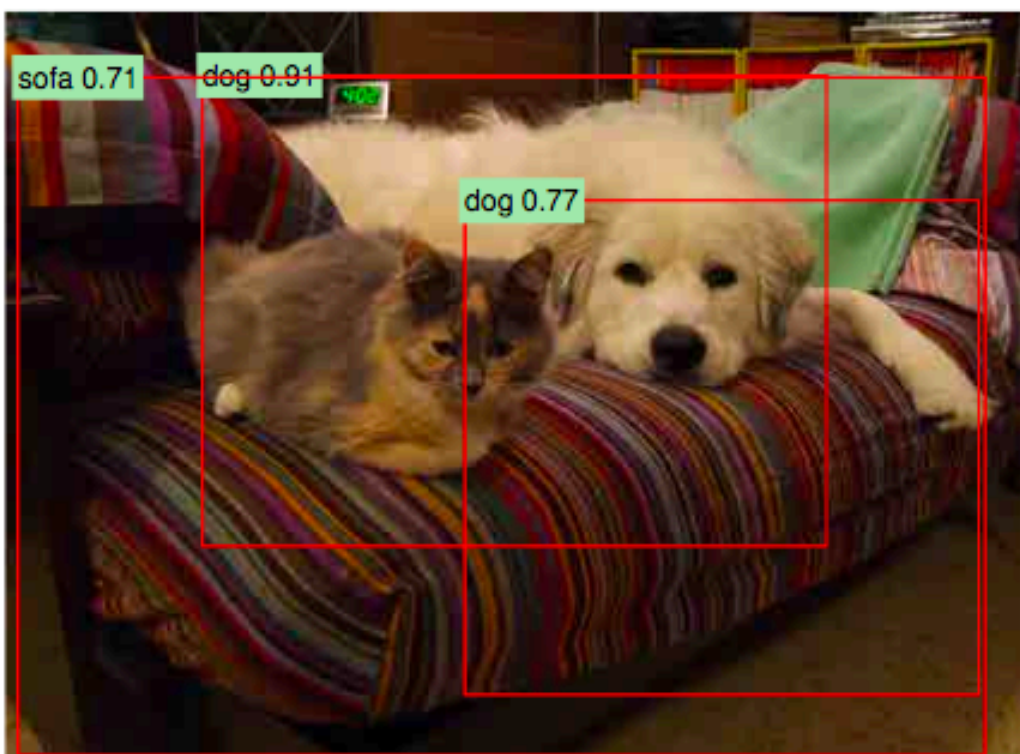
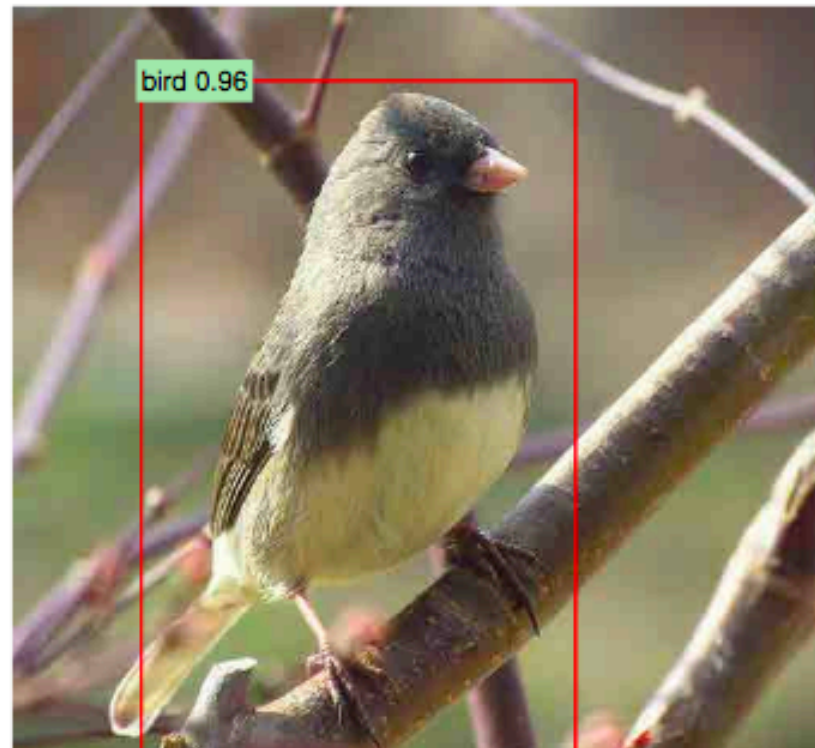
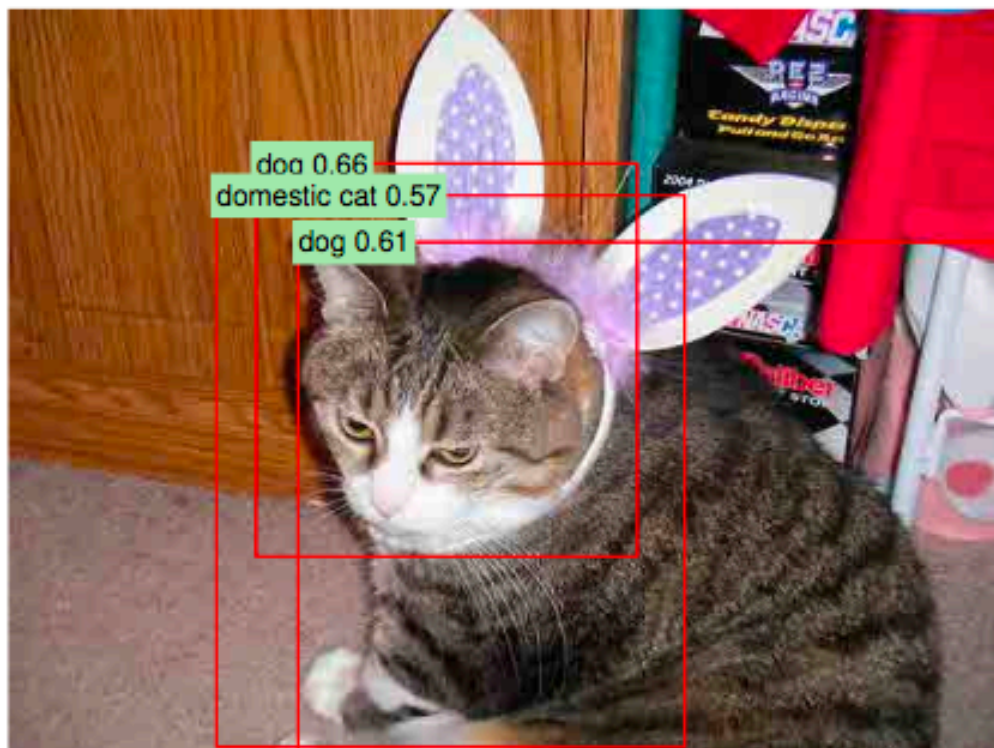
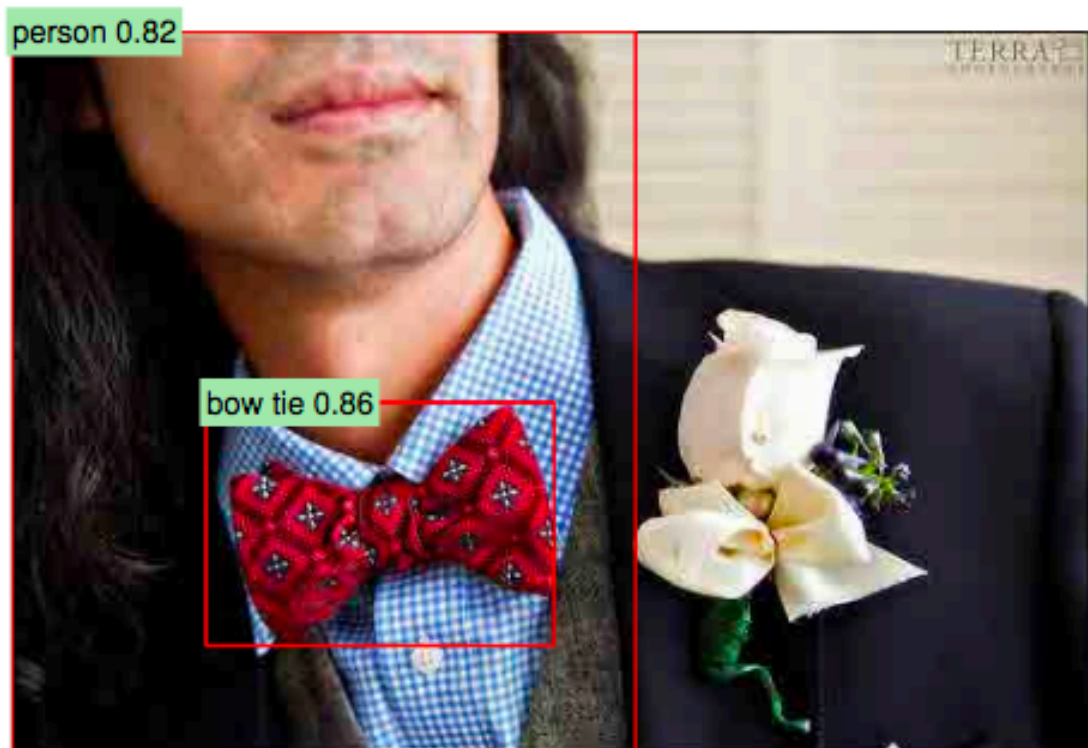
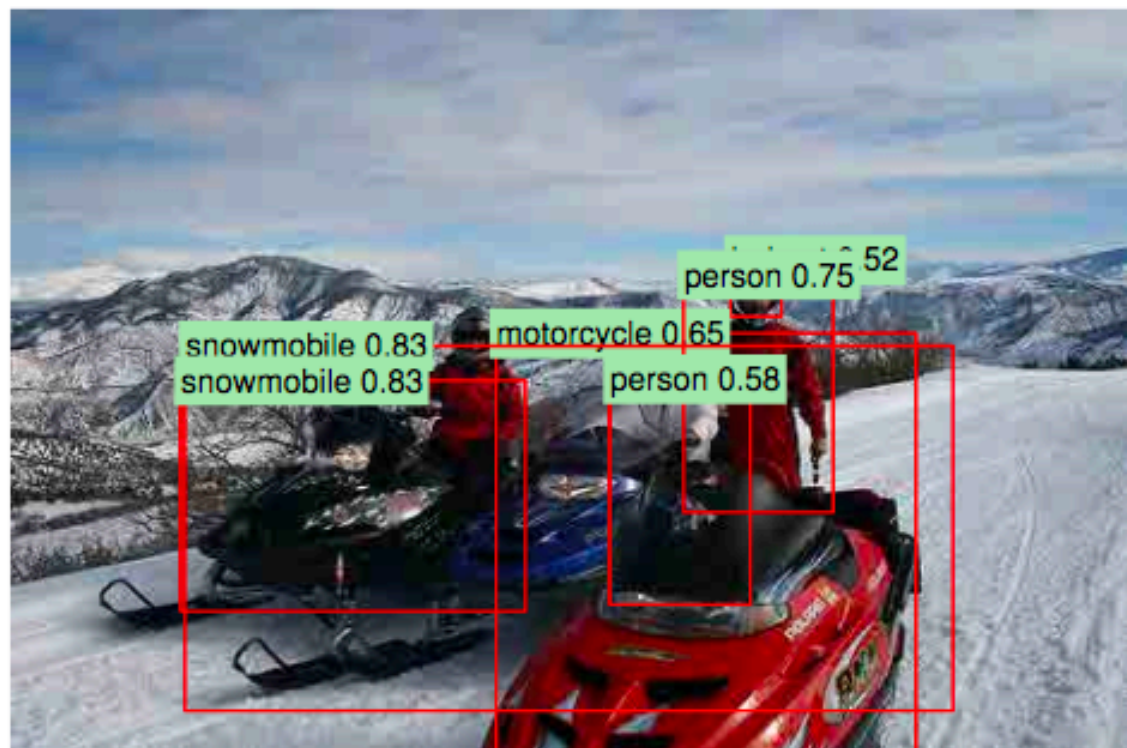
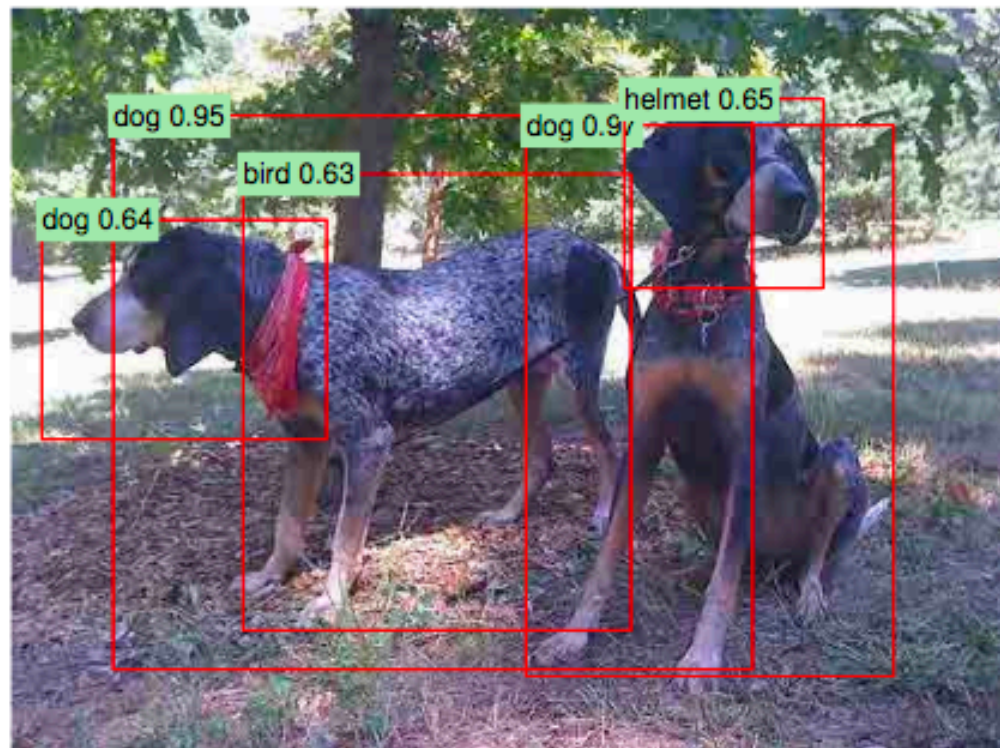
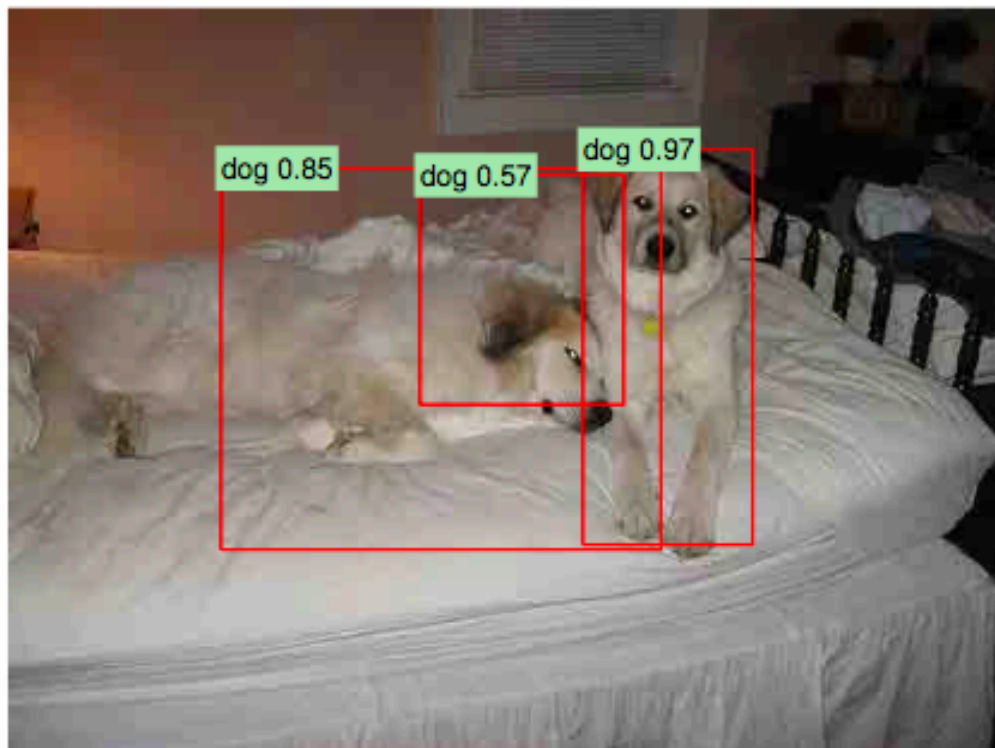
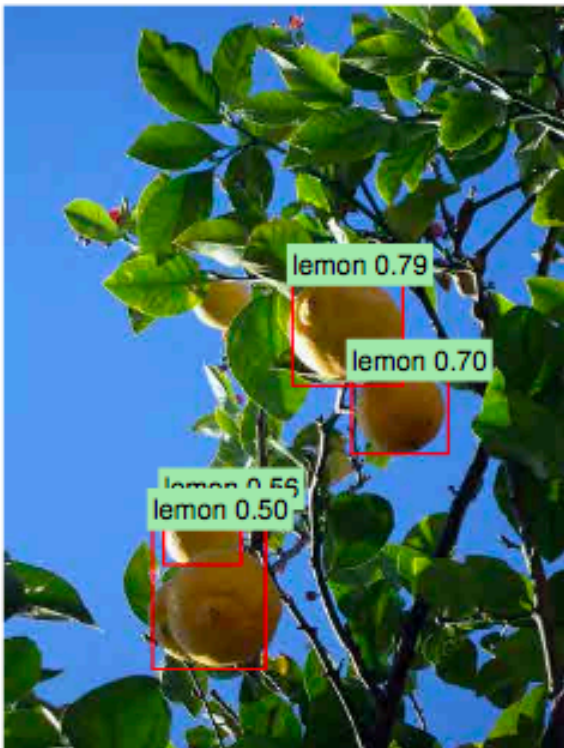
R-CNN

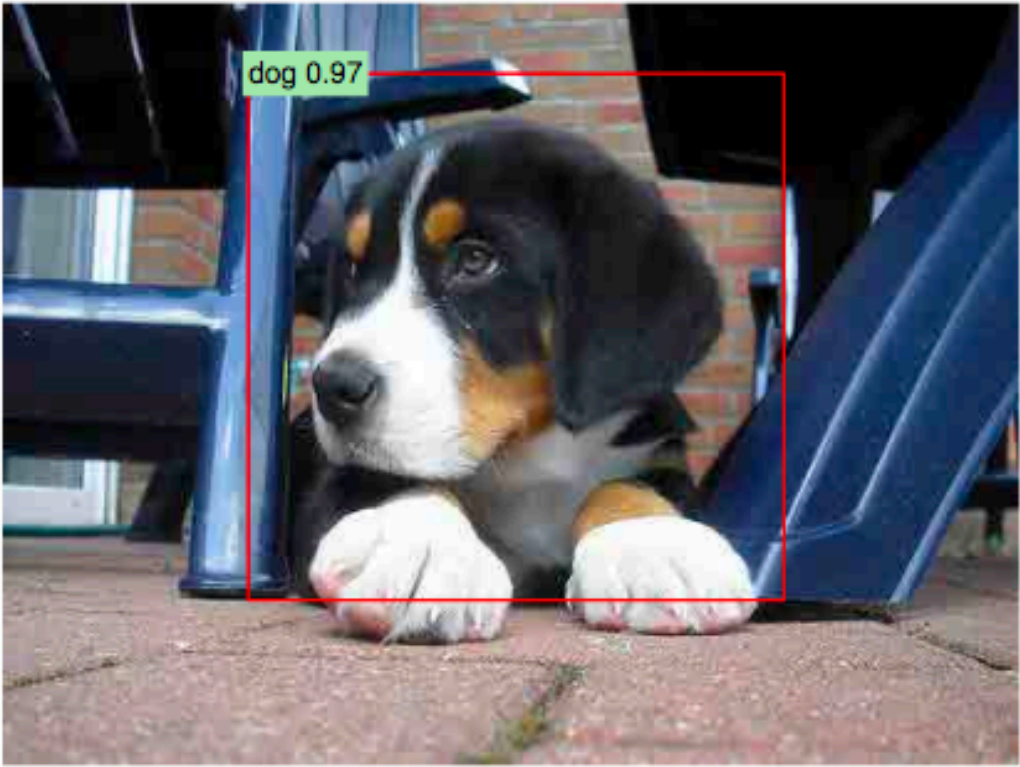
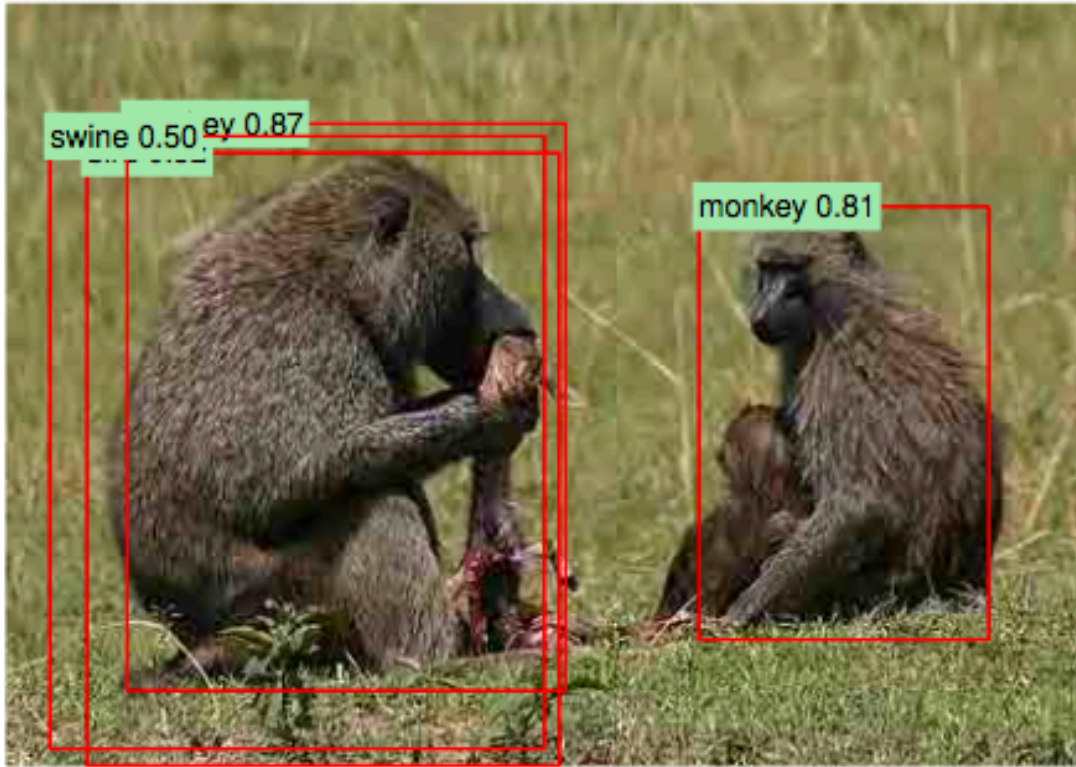
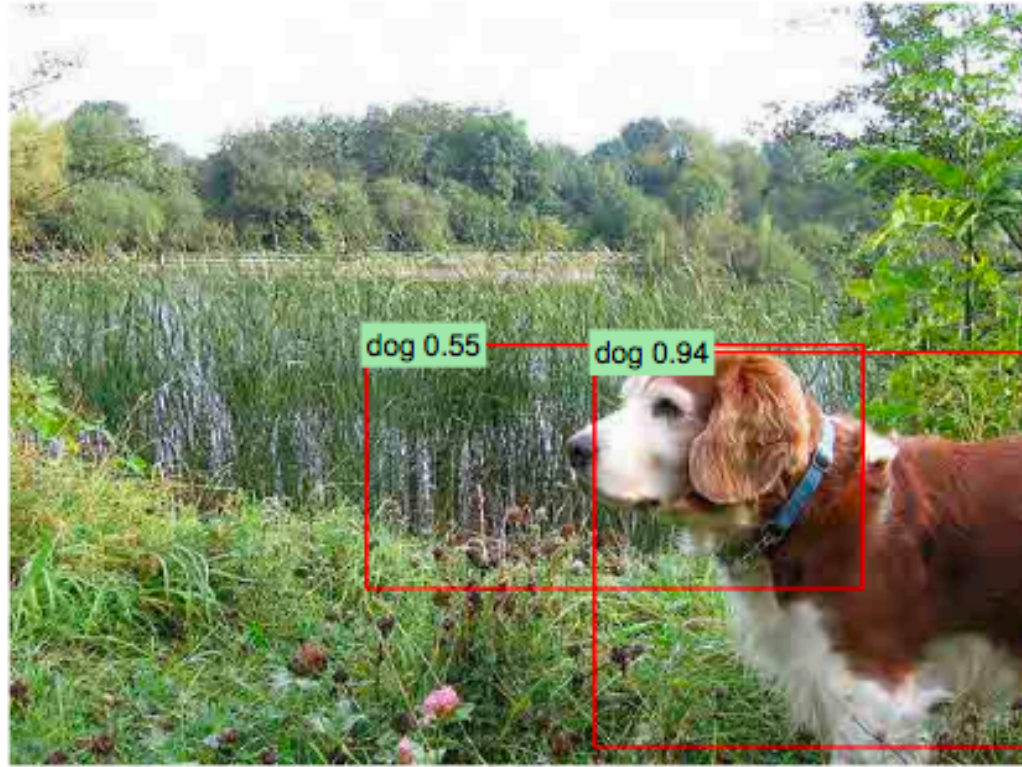
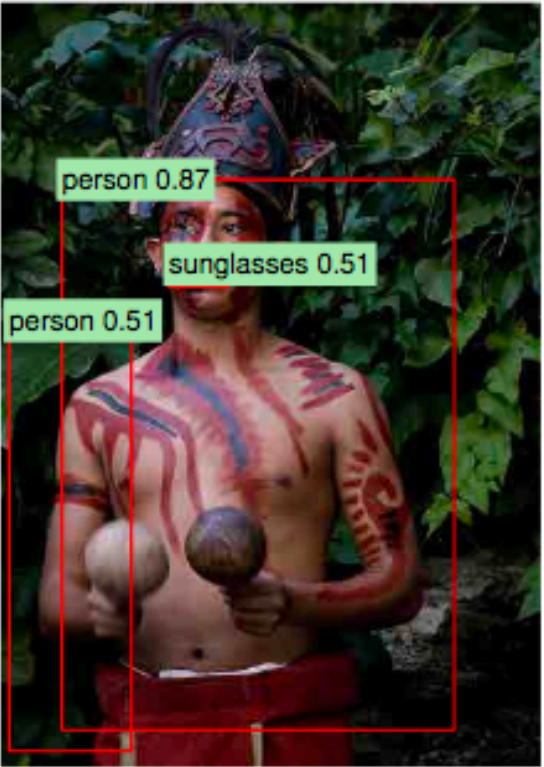
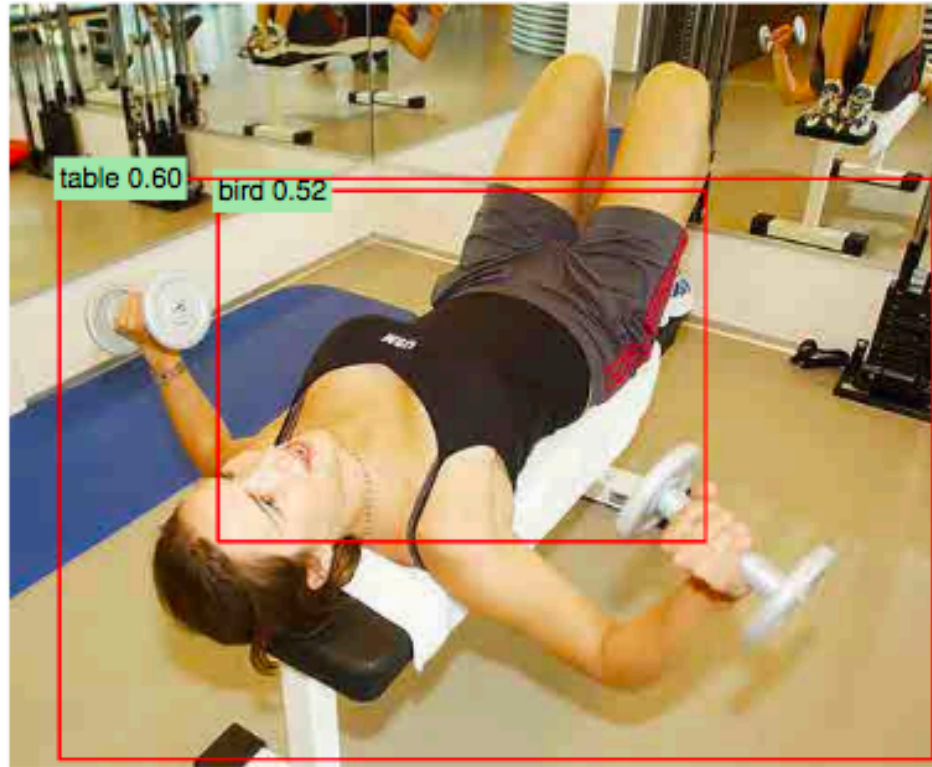
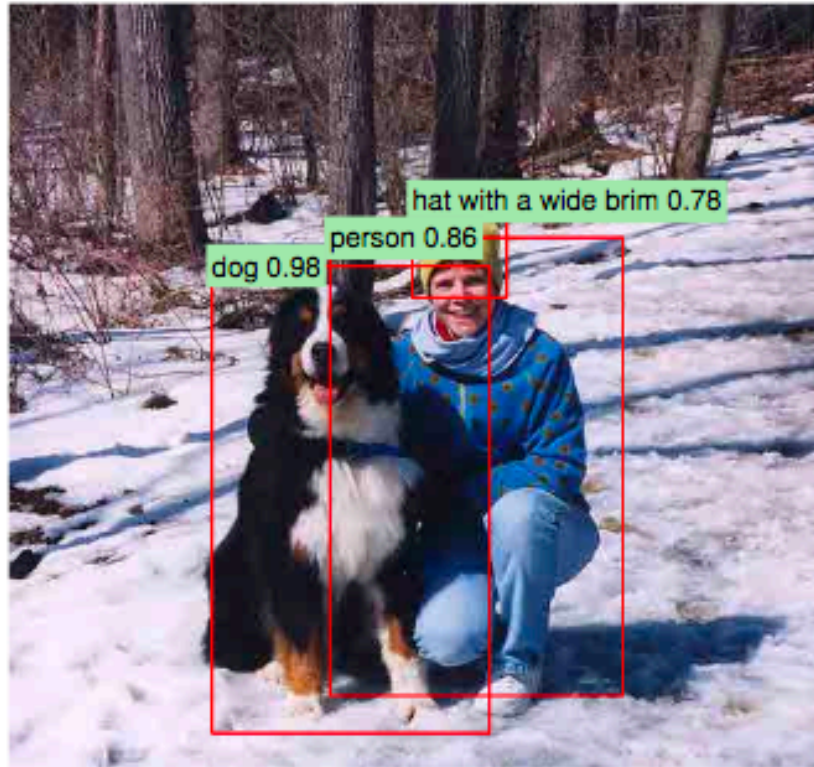


R-CNN

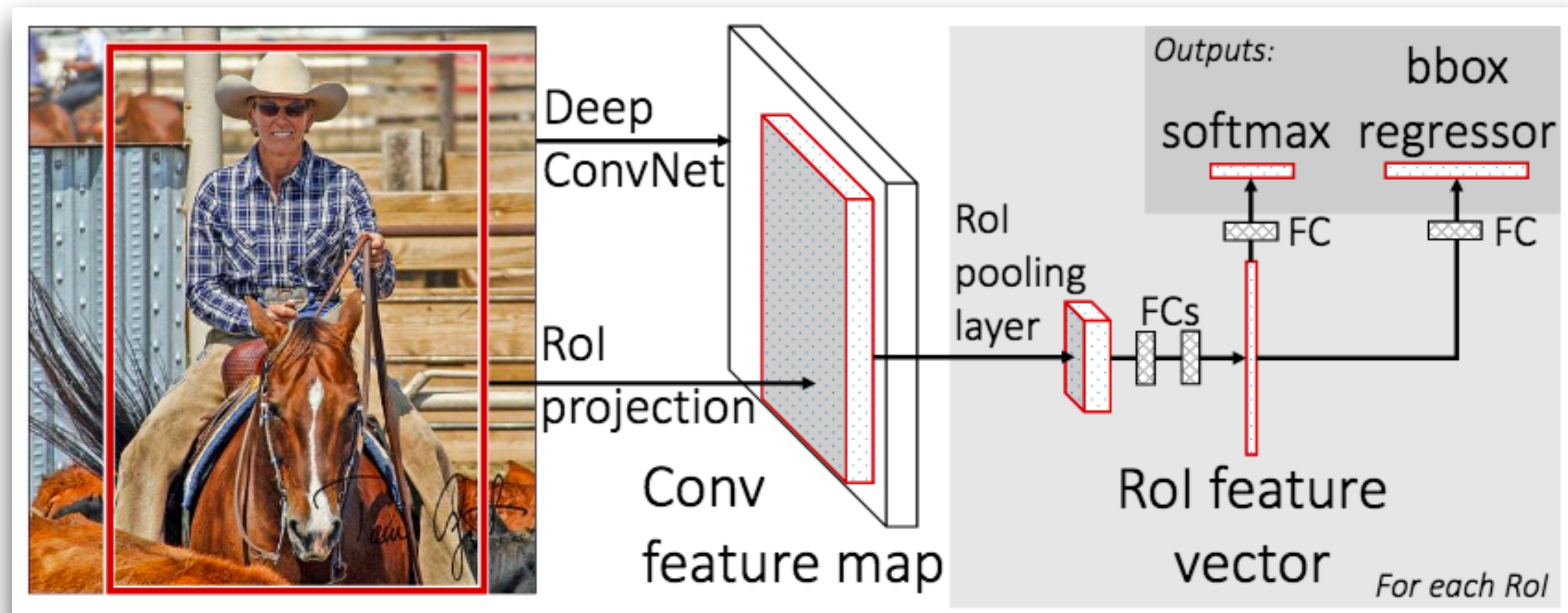


Girshick, R., J. Donahue, T. Darrell, and J. Malik. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." *CVPR '14 Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*. Pages 580-587. 2014



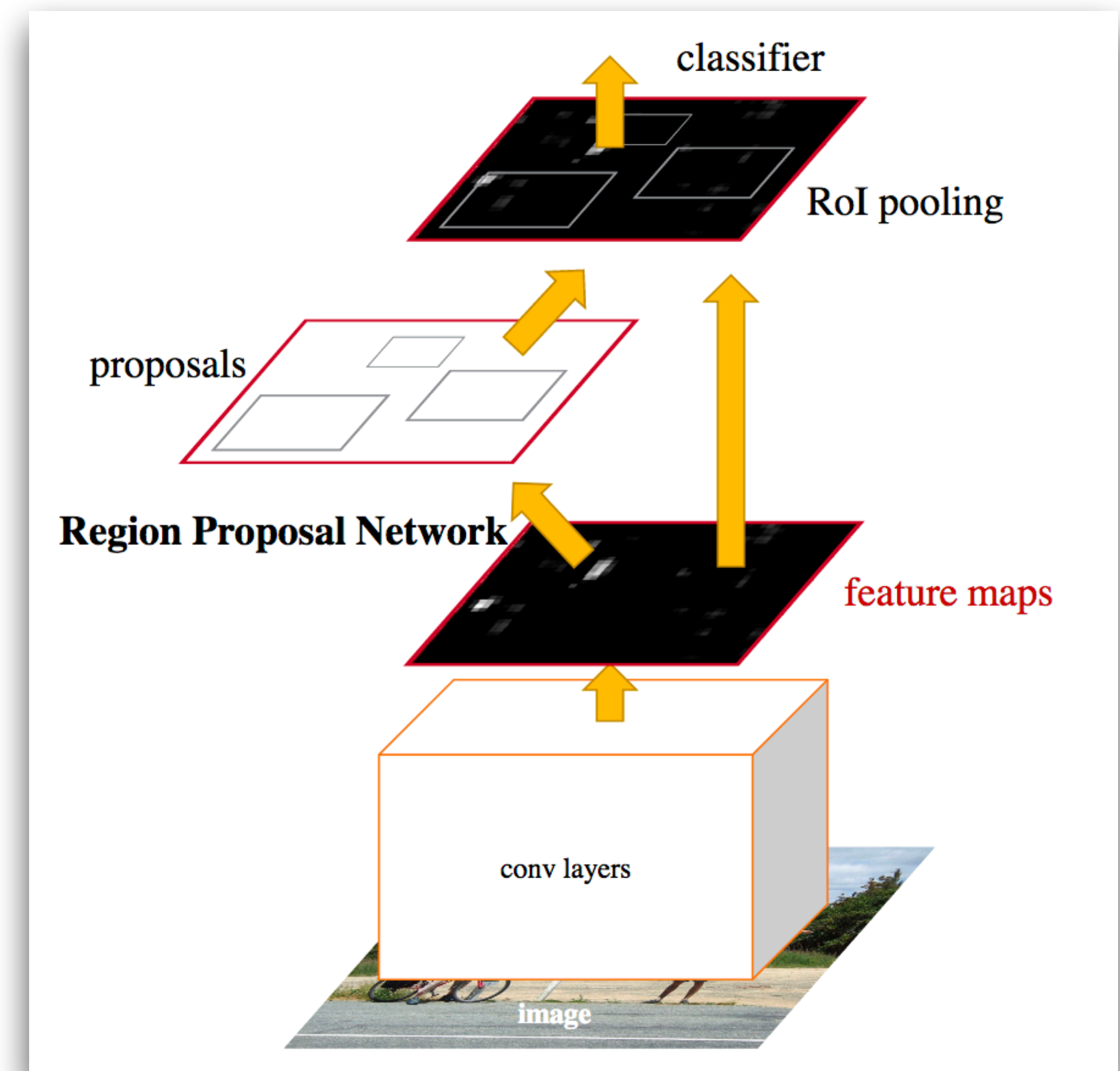


Making the structure end-to-end



<https://arxiv.org/pdf/1504.08083.pdf>

Fast R-CNN

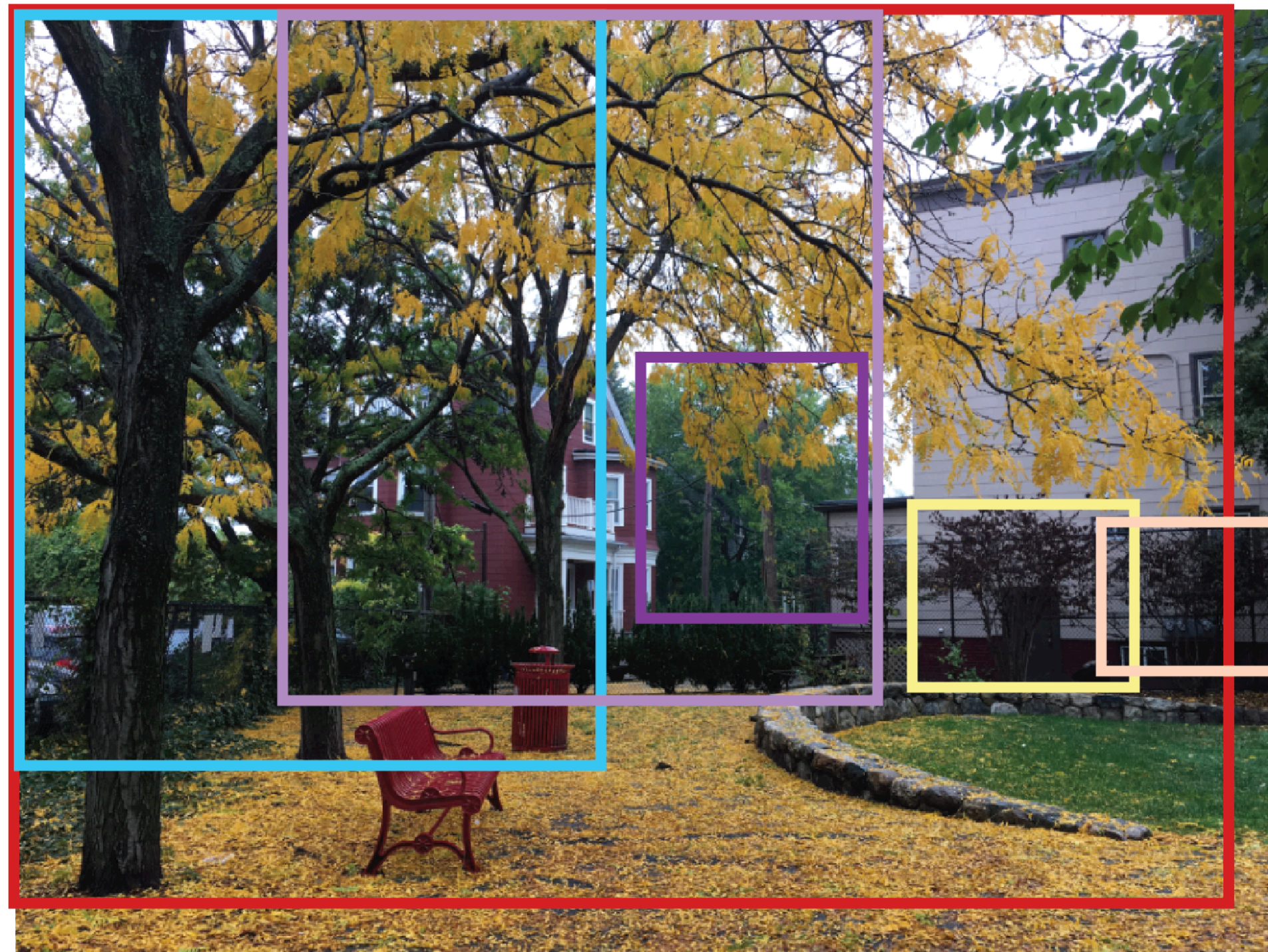
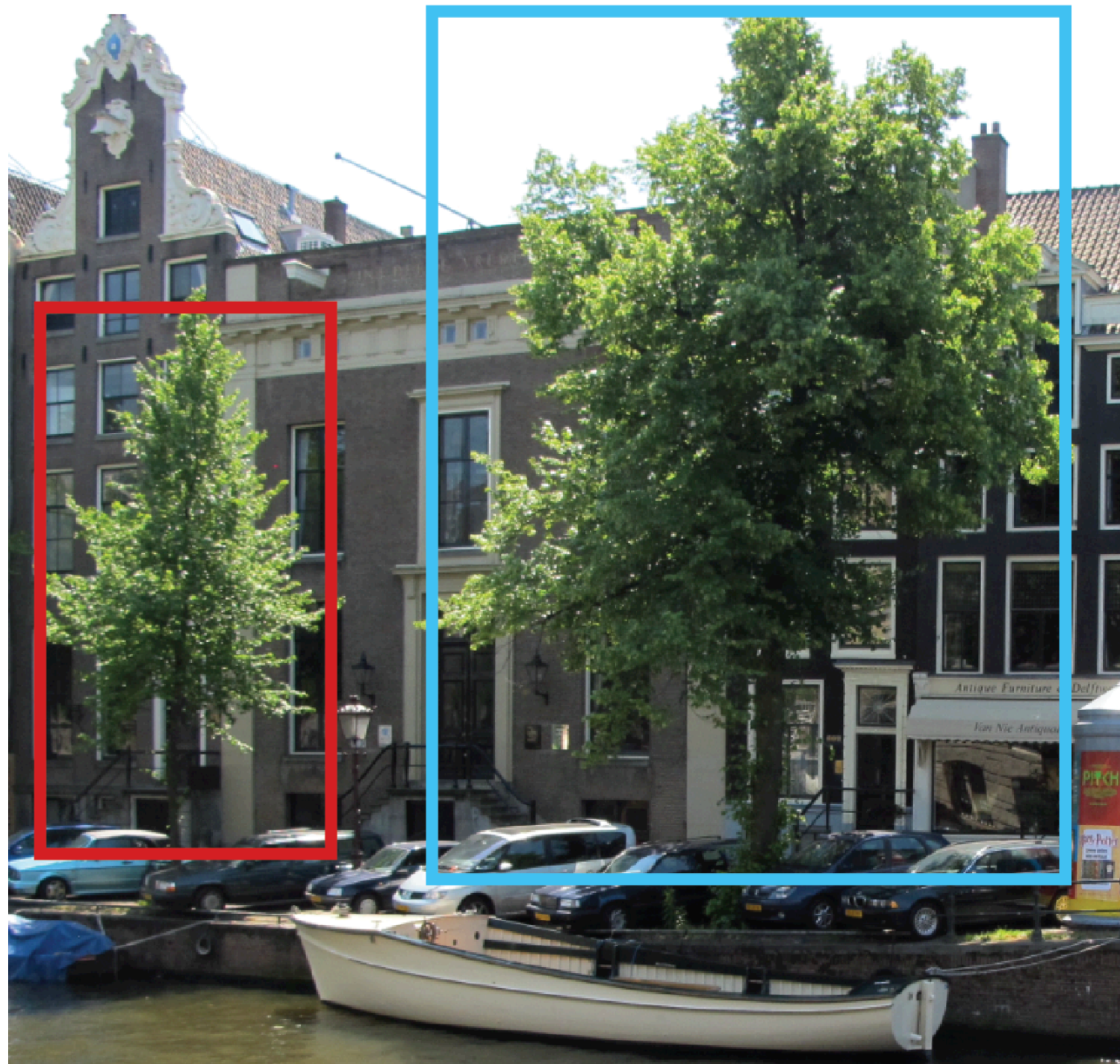


<https://arxiv.org/pdf/1506.01497.pdf>

Faster R-CNN

Bounding box localization: shortcomings

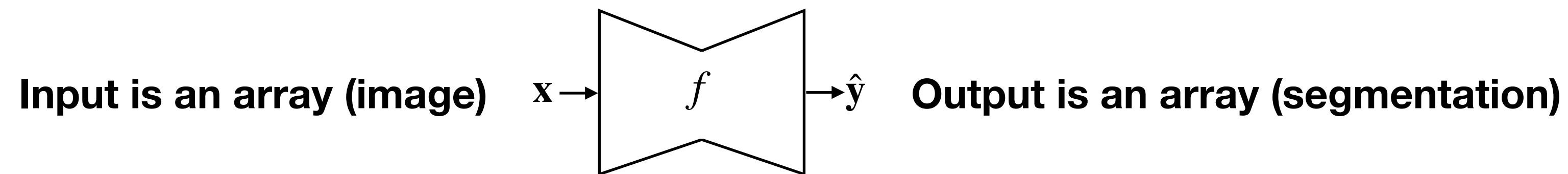
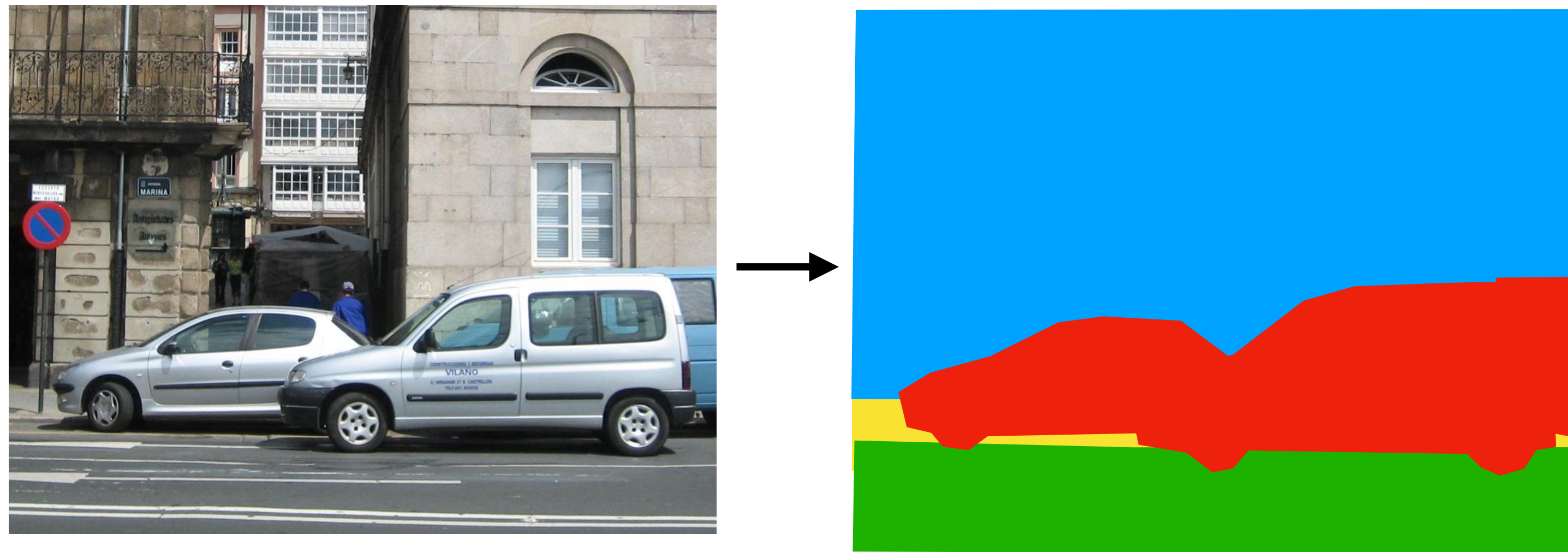
It inherits all the problems of classification plus:

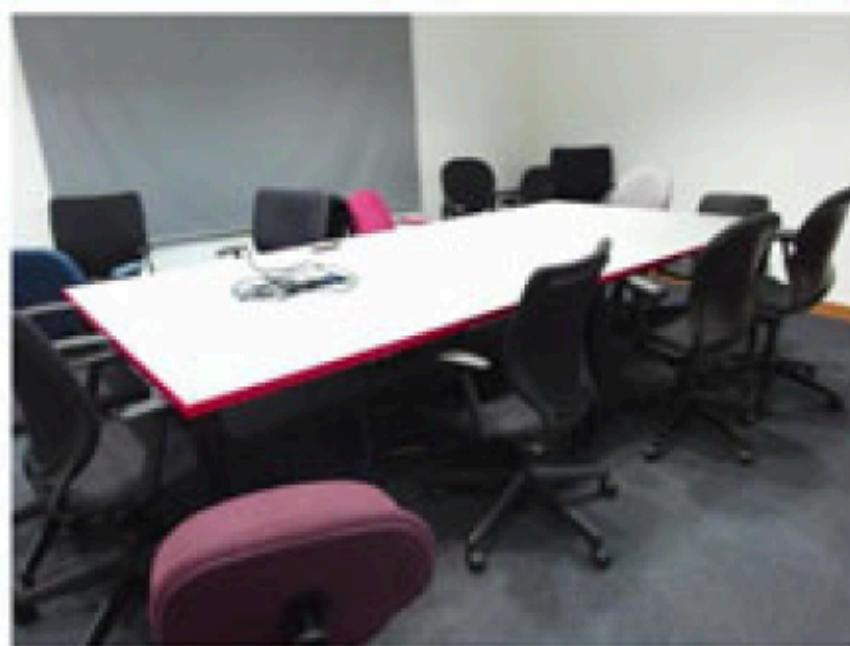
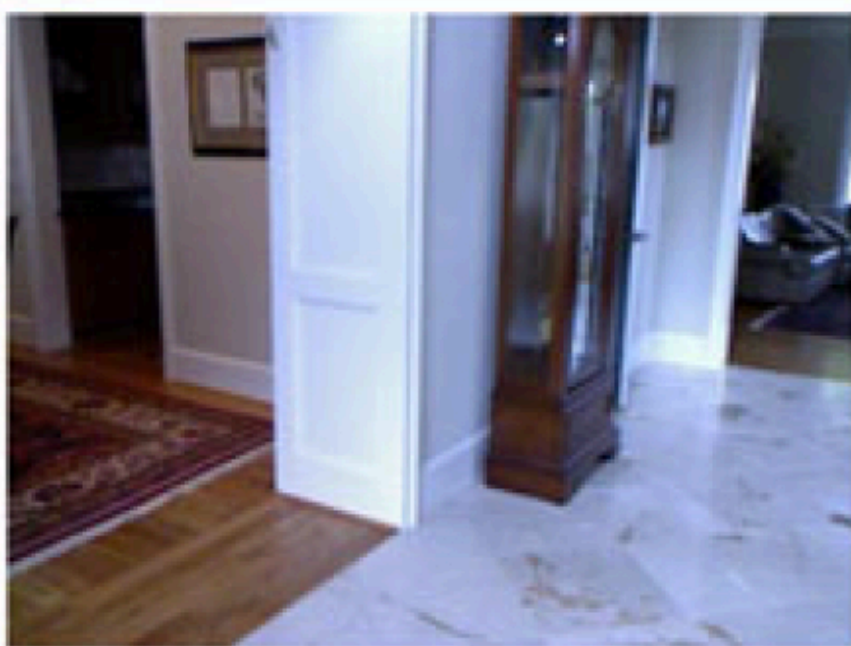
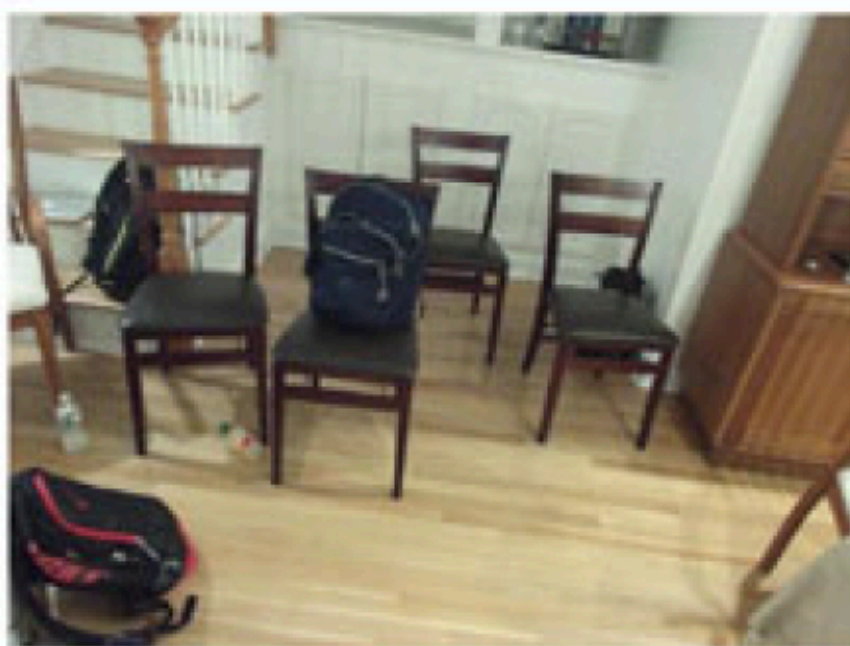
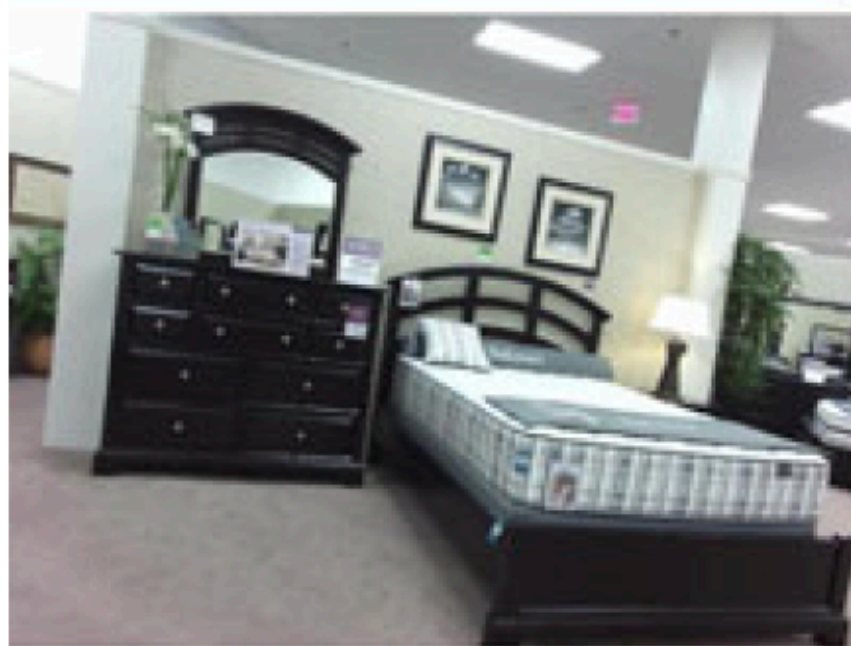
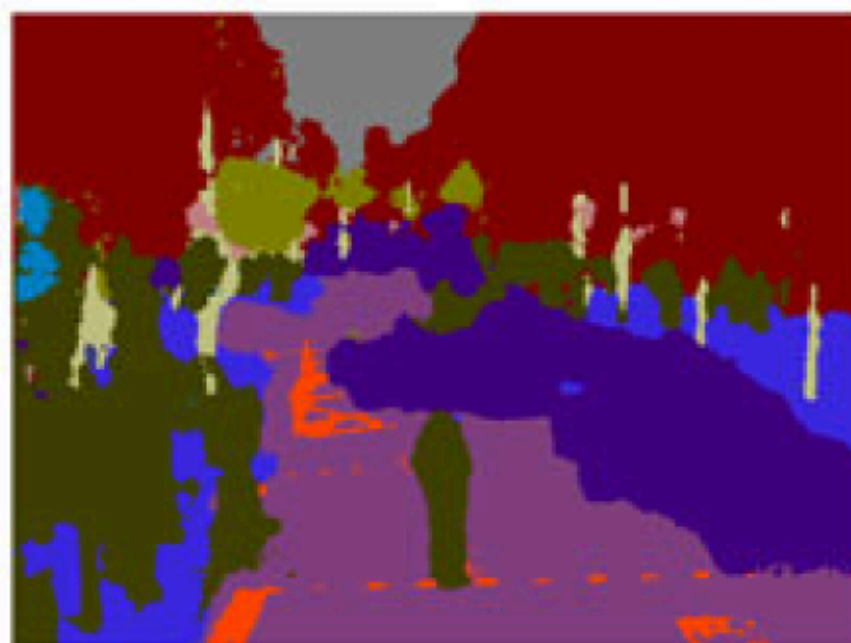
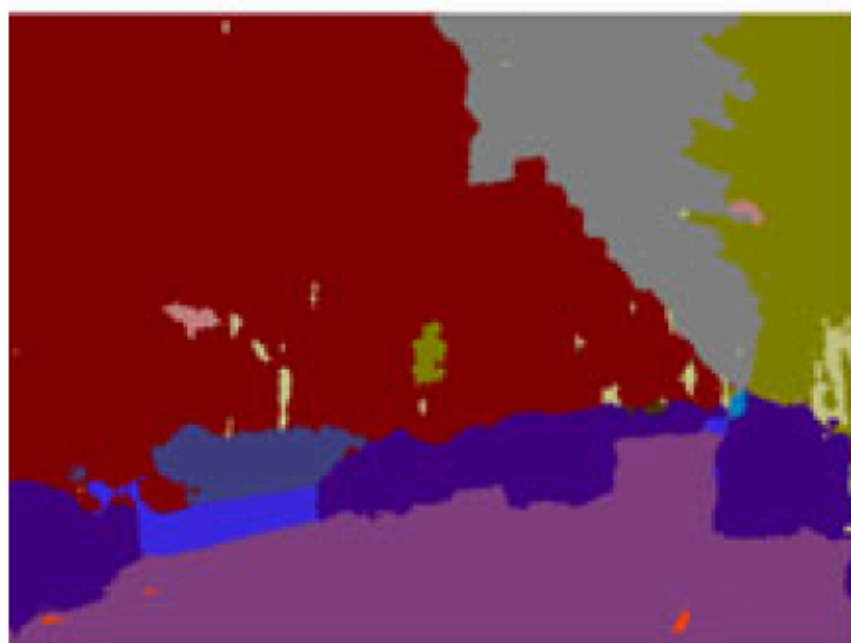
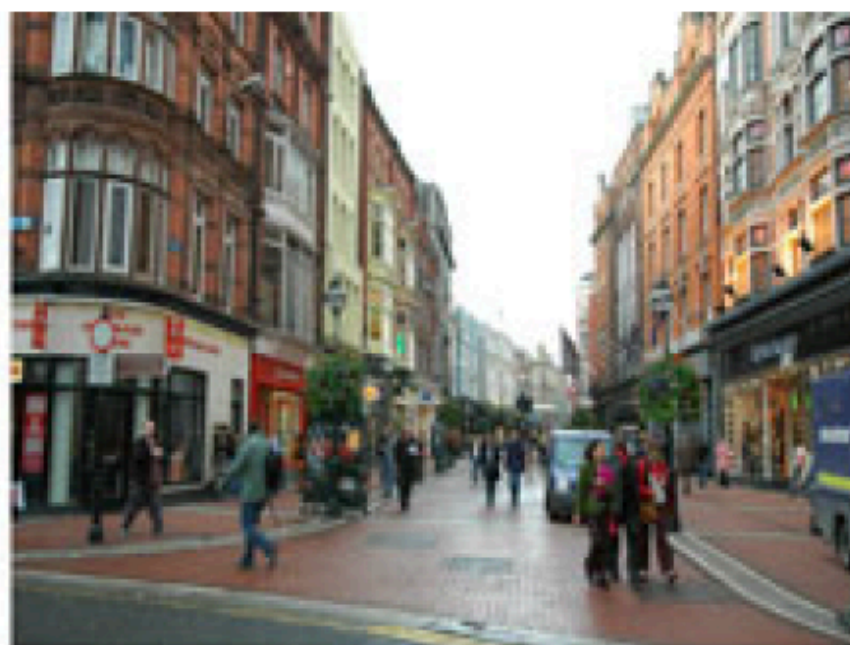
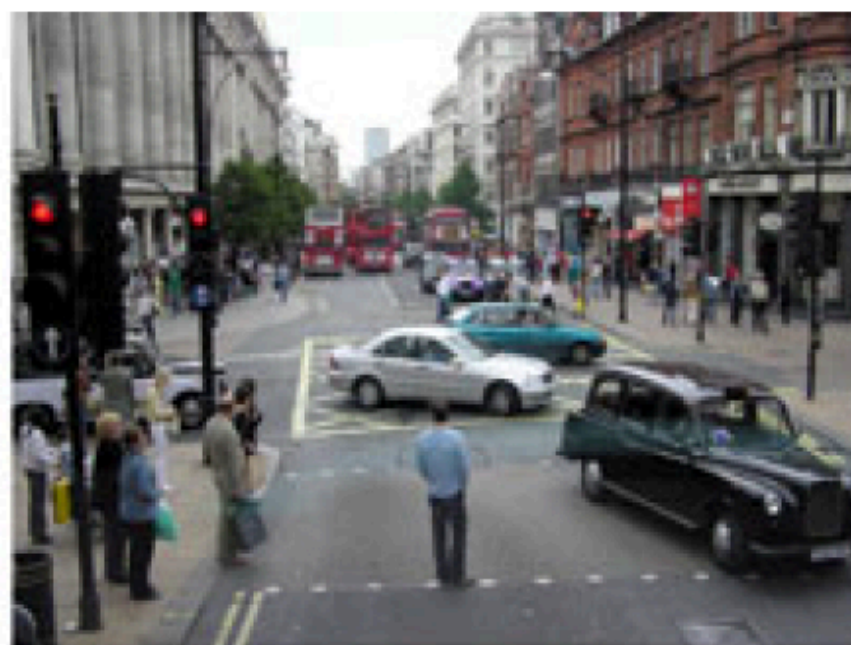
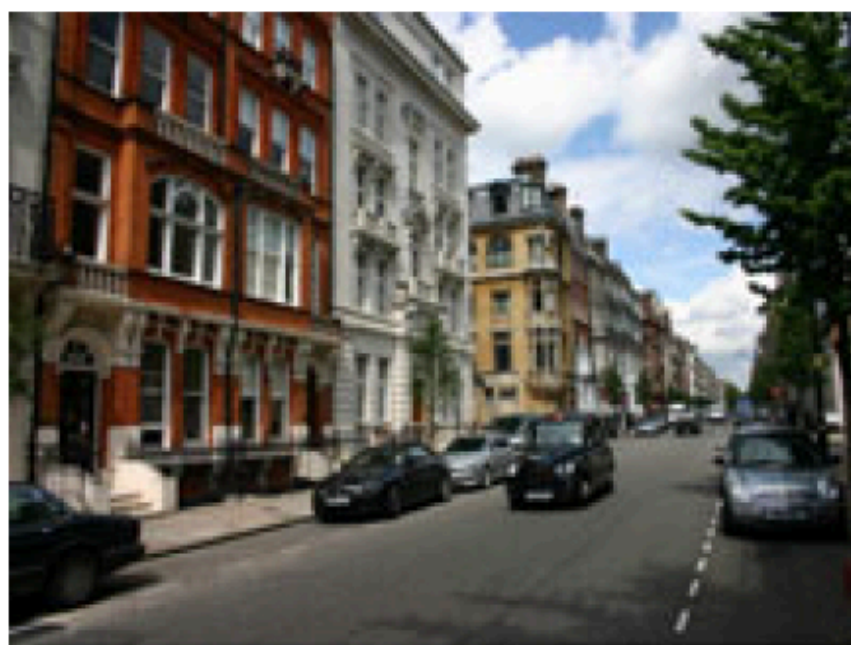


Object segmentation

We can try to classify each pixel in an image with an object class. Per-pixel classification of object labels is referred to as semantic segmentation.

Is object class c present in the pixel $[n,m]$?

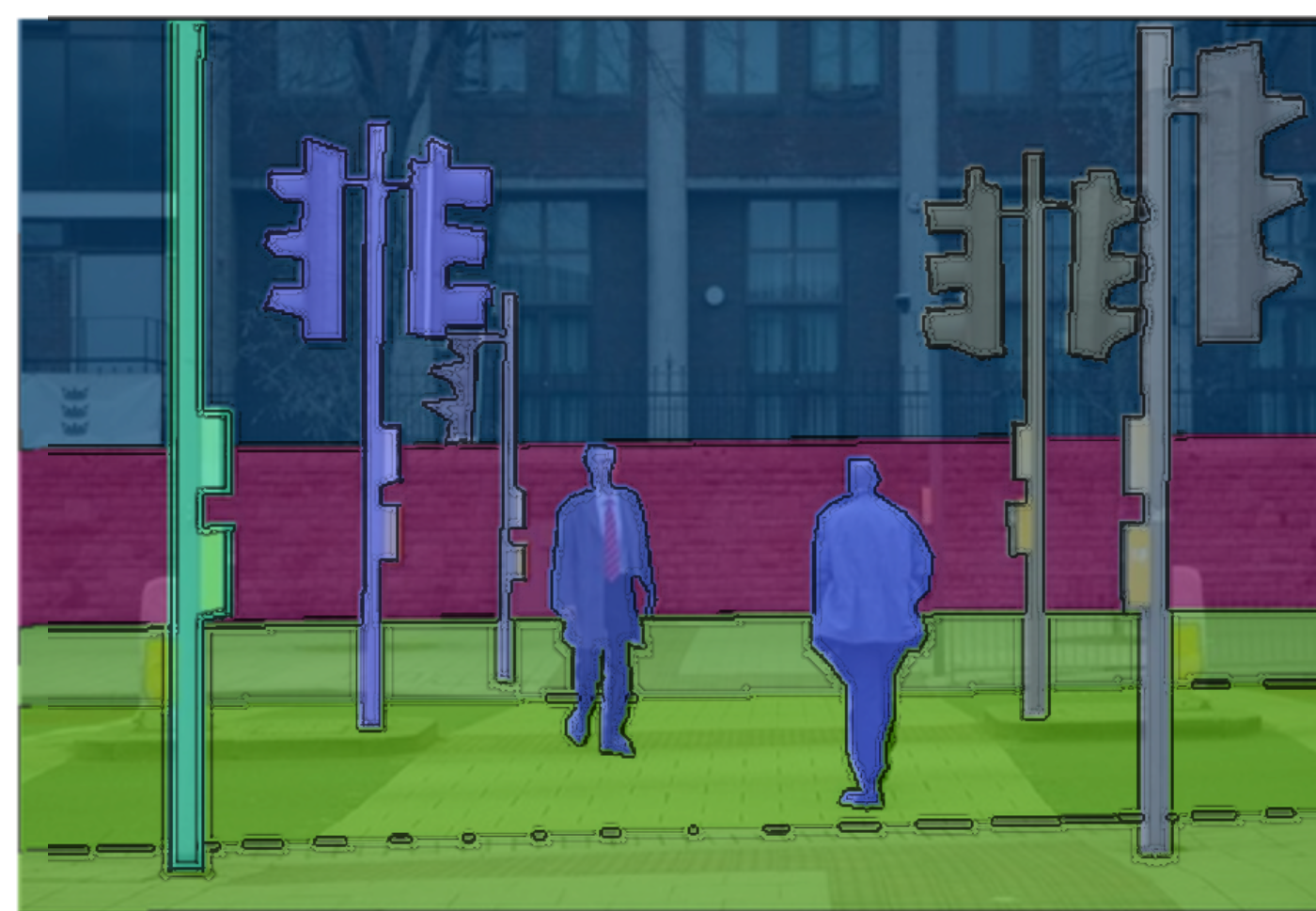




COCO



ADE20K

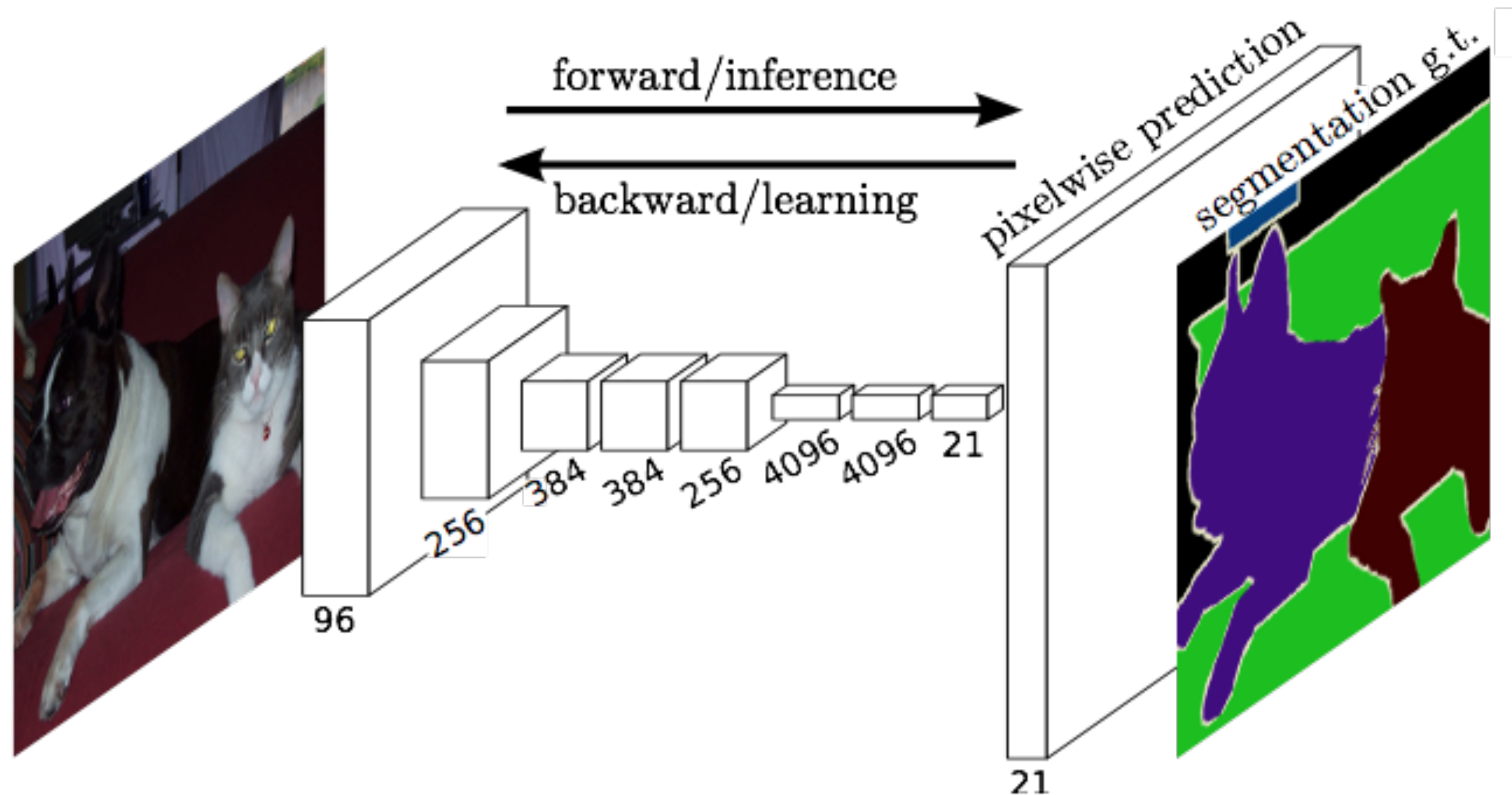


	Images	Obj. inst.	Obj. classes	Part inst.	Part classes	Obj. classes per image
COCO	123,287	886,284	91	0	0	3.5
ImageNet*	476,688	534,309	200	0	0	1.7
NYU Depth V2	1,449	34,064	894	0	0	14.1
Cityscapes	25,000	N/A	30	0	0	N/A
SUN	16,873	313,884	4,479	0	0	9.8
OpenSurfaces	22,214	71,460	160	0	0	N/A
PascalContext	10,103	~104,398**	540	181,770	40	5.1
ADE20K	22,000	415,099	2,944	171,148	354	10.5

* has only bounding boxes (no pixel-level segmentation). Sparse annotations.

** PascalContext dataset does not have instance segmentation. In order to estimate the number of instances, we find connected components (having at least 150pixels) for each class label.

Fully Convolutional Networks



Fully Convolutional Networks for Semantic Segmentation

Jonathan Long* Evan Shelhamer* Trevor Darrell
UC Berkeley
{jonlong, shelhamer, trevor}@cs.berkeley.edu

Abstract

Convolutional networks are powerful visual models that yield hierarchies of features. We show that convolutional networks by themselves, trained end-to-end, pixels-to-pixels, exceed the state-of-the-art in semantic segmentation. Our key insight is to build “fully convolutional” networks that take input of arbitrary size and produce correspondingly-sized output with efficient inference and learning. We define and detail the space of fully convolutional networks, explain their application to spatially dense prediction tasks, and draw connections to prior models. We adapt contemporary classification networks (AlexNet [22], the VGG net [34], and GoogLeNet [35]) into fully convolutional networks and transfer their learned representations by fine-tuning [5] to the segmentation task. We then define a skip architecture that combines semantic information from a deep, coarse layer with appearance information from a shallow, fine layer to produce accurate and detailed segmentations. Our fully convolutional network achieves state-of-the-art segmentation of PASCAL VOC (20% relative improvement to 62.2% mean IU on 2012), NYUDv2, and SIFT Flow, while inference takes less than one fifth of a second for a typical image.

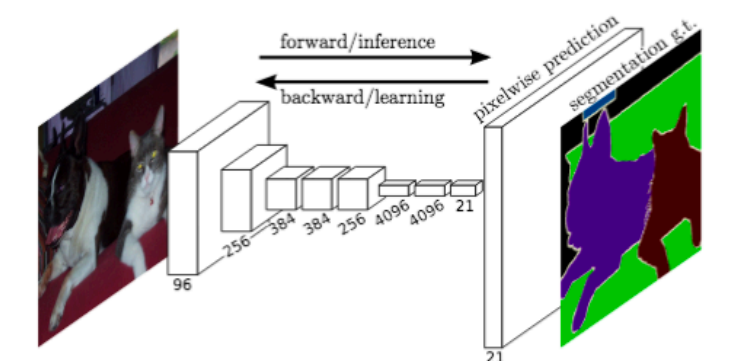


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

We show that a fully convolutional network (FCN) trained end-to-end, pixels-to-pixels on semantic segmentation exceeds the state-of-the-art without further machinery. To our knowledge, this is the first work to train FCNs end-to-end (1) for pixelwise prediction and (2) from supervised pre-training. Fully convolutional versions of existing networks predict dense outputs from arbitrary-sized inputs. Both learning and inference are performed whole-image-at-a-time by dense feedforward computation and backpropagation. In-network upsampling layers enable pixelwise prediction and learning in nets with subsampled pooling.

This method is efficient, both asymptotically and absolutely, and precludes the need for the complications in other works. Patchwise training is common [30, 3, 9, 31, 11], but lacks the efficiency of fully convolutional training. Our approach does not make use of pre- and post-processing complications, including superpixels [9, 17], proposals [17, 15], or post-hoc refinement by random fields or local classifiers [9, 17]. Our model transfers recent success in classification [22, 34, 35] to dense prediction by reinterpreting classification nets as fully convolutional and fine-tuning from their learned representations. In contrast, previous works have applied small convnets without supervised pre-training [9, 31, 30].

Semantic segmentation faces an inherent tension between semantics and location: global information resolves what while local information resolves where. Deep feature hierarchies encode location and semantics in a nonlinear

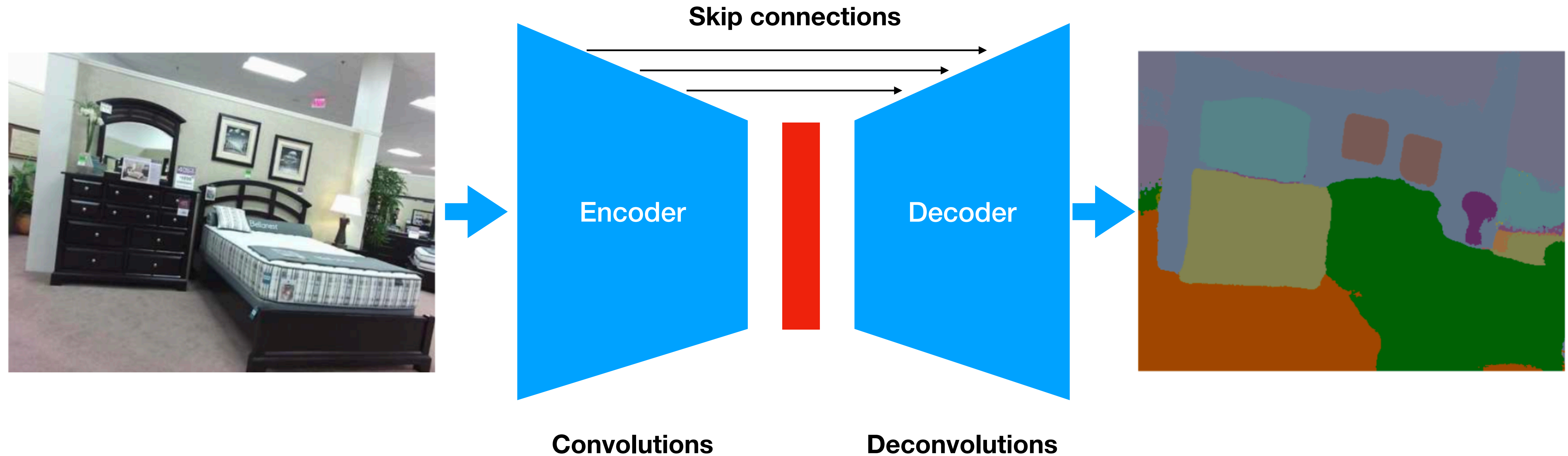
1. Introduction

Convolutional networks are driving advances in recognition. Convnets are not only improving for whole-image classification [22, 34, 35], but also making progress on local tasks with structured output. These include advances in bounding box object detection [32, 12, 19], part and key-point prediction [42, 26], and local correspondence [26, 10].

The natural next step in the progression from coarse to fine inference is to make a prediction at every pixel. Prior approaches have used convnets for semantic segmentation [30, 3, 9, 31, 17, 15, 11], in which each pixel is labeled with the class of its enclosing object or region, but with shortcomings that this work addresses.

* Authors contributed equally

Encoder-decoder architectures



Encoder-decoder architectures

SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation

R. Cipolla, Senior Member, IEEE,

This paper presents a deep convolutional encoder-decoder architecture for semantic pixel-wise segmentation. The encoder network, a corresponding decoder network followed by a skip connection, is topologically identical to the 13 convolutional layers in the VGG16 network. The decoder network upsamples its lower resolution input feature maps to the resolution of the input image. The upsampling is performed by max-pooling step of the corresponding encoder to produce sparse maps. The upsampling maps are sparse and are then combined with the corresponding encoder feature maps to produce the final segmentation map. This comparison reveals the memory versus

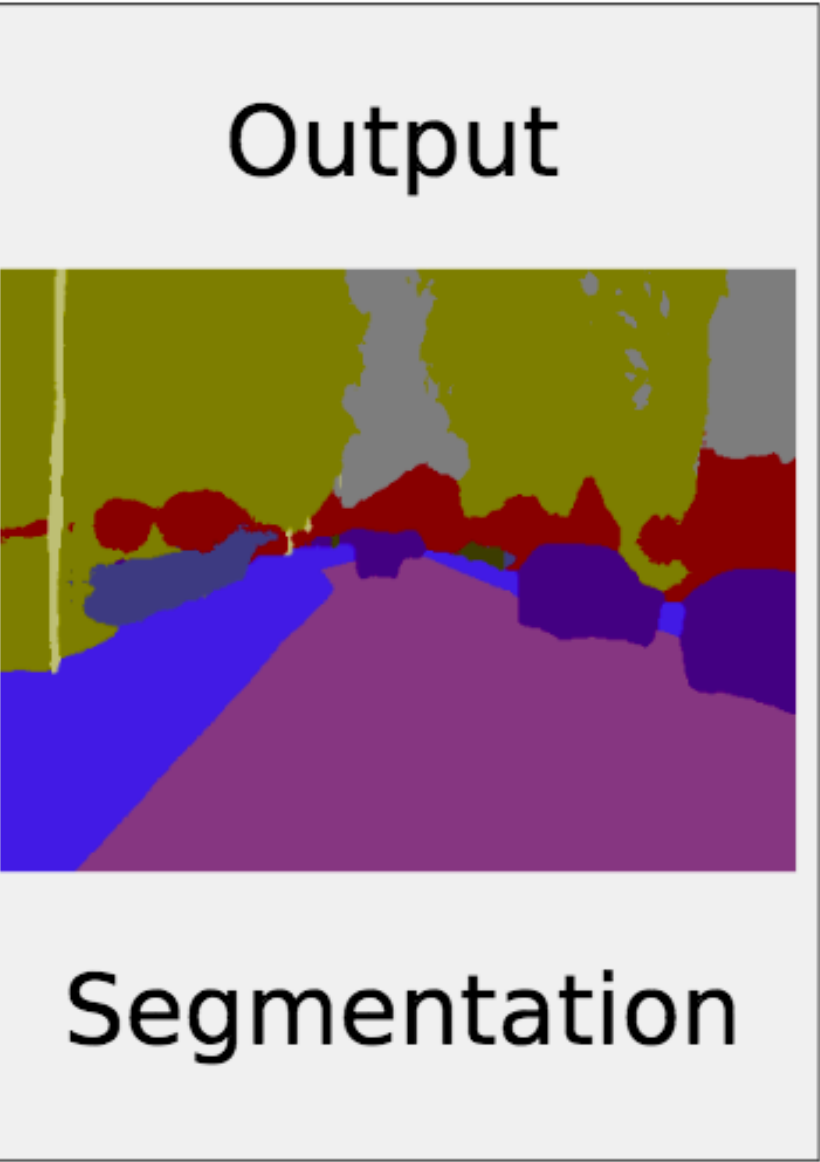
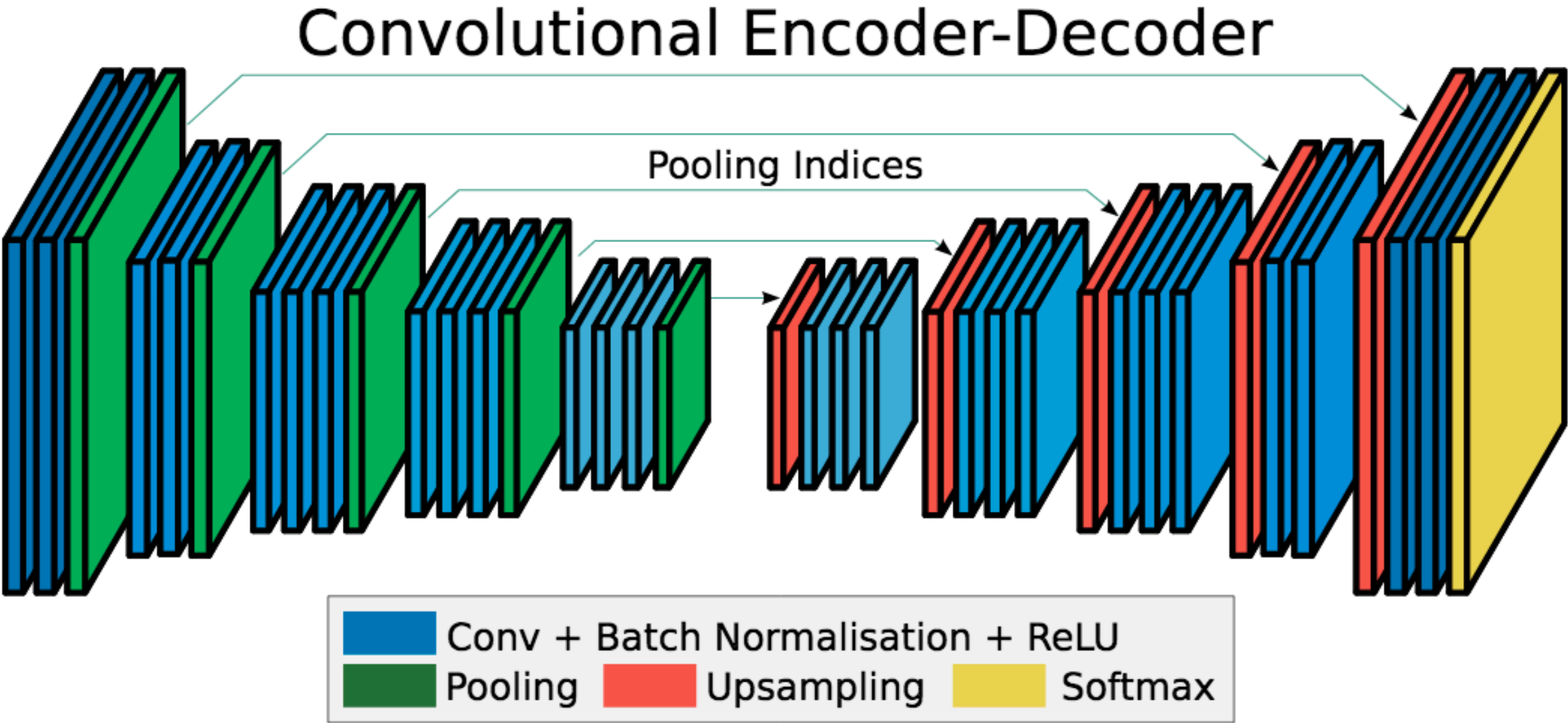
performance of the proposed architecture with the widely adopted FCN [2]. This comparison reveals the memory versus

segmentation, Indoor Scenes, Road Scenes, Encoder,

ns) and understand the spatial-relationship (context) between different classes such as road and side-walk. In typical road scene images, the majority of the pixels belong to large classes such as building and hence the network must produce smooth boundaries. The engine must also have the ability to delineate boundaries based on their shape despite their small size. Hence it is necessary to retain boundary information in the extracted image features. From a computational perspective, it is necessary for the network to be efficient in terms of both memory and inference time during inference. The ability to train end-to-end to jointly optimise all the weights in the network using a single weight update technique such as stochastic gradient descent (SGD) [17] is an additional benefit since it is more easily repeatable. The design of SegNet arose from a need to match these criteria.

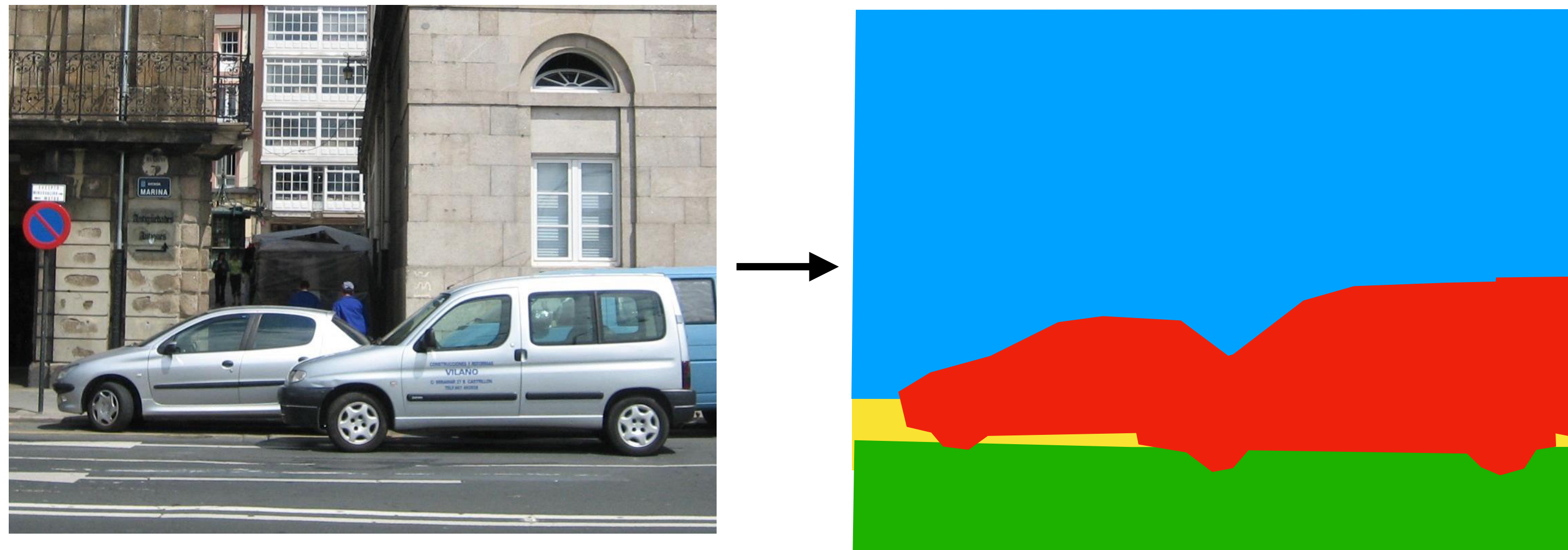
The encoder network in SegNet is topologically identical to the convolutional layers in VGG16 [1]. We remove the fully connected layers of VGG16 which makes the SegNet encoder network significantly smaller and easier to train than many other recent architectures [2], [4], [11], [18]. The key component of SegNet is the decoder network which consists of a hierarchy of decoders one corresponding to each encoder. Of these, the appropriate decoders use the max-pooling indices received from the corresponding encoder to perform non-linear upsampling of their input feature maps. This idea was inspired from an architecture designed for unsupervised feature learning [19]. Reusing max-pooling indices in the decoding process has several practical

• V. Badrinarayanan, A. Kendall, R. Cipolla are with the Machine Intelligence Lab, Department of Engineering, University of Cambridge, UK. E-mail: vb292,agk34,cipolla@eng.cam.ac.uk



Object segmentation: shortcomings

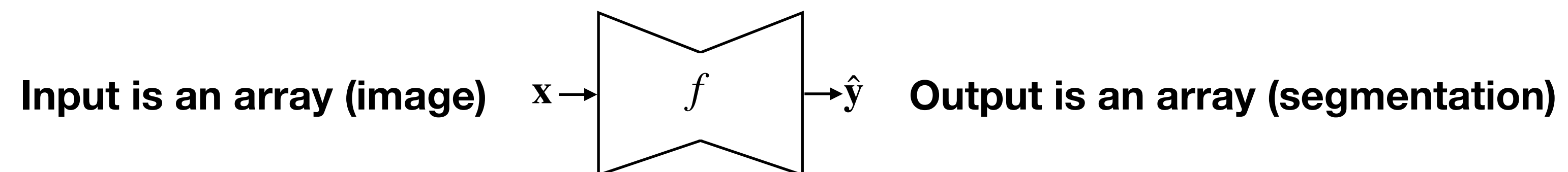
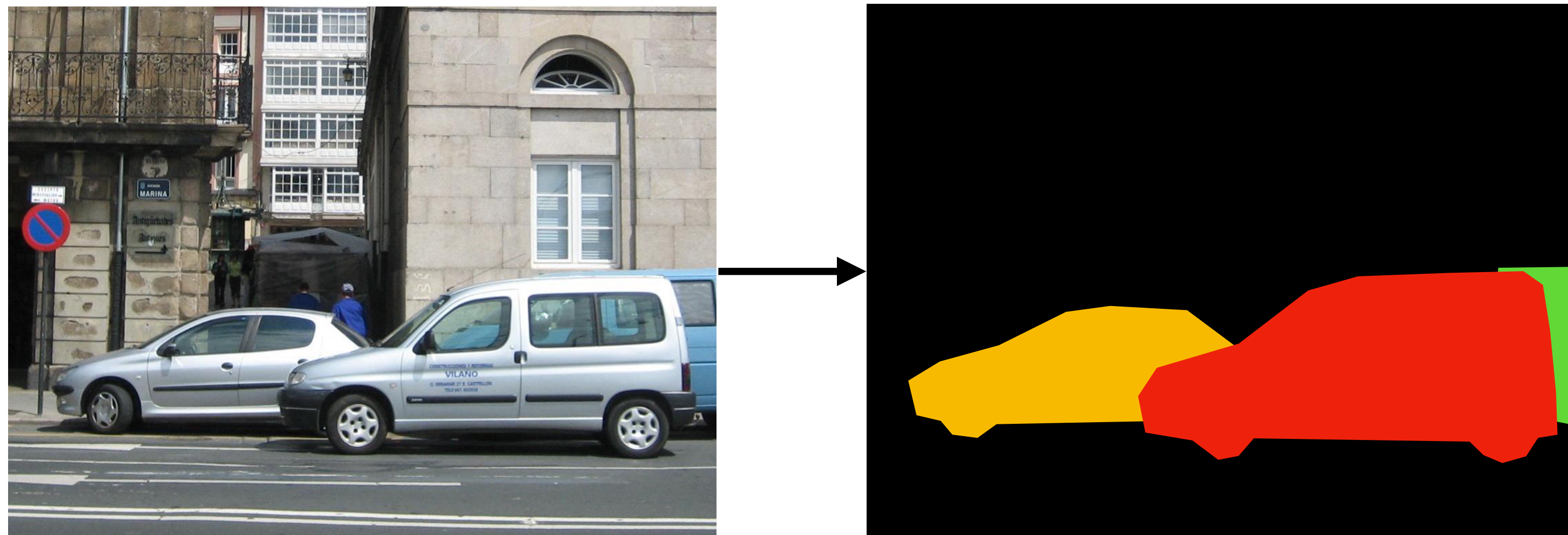
We can not count objects!



Instance segmentation

We can try to classify each pixel in an image with an object class. Per-pixel classification of object labels is referred to as semantic segmentation.

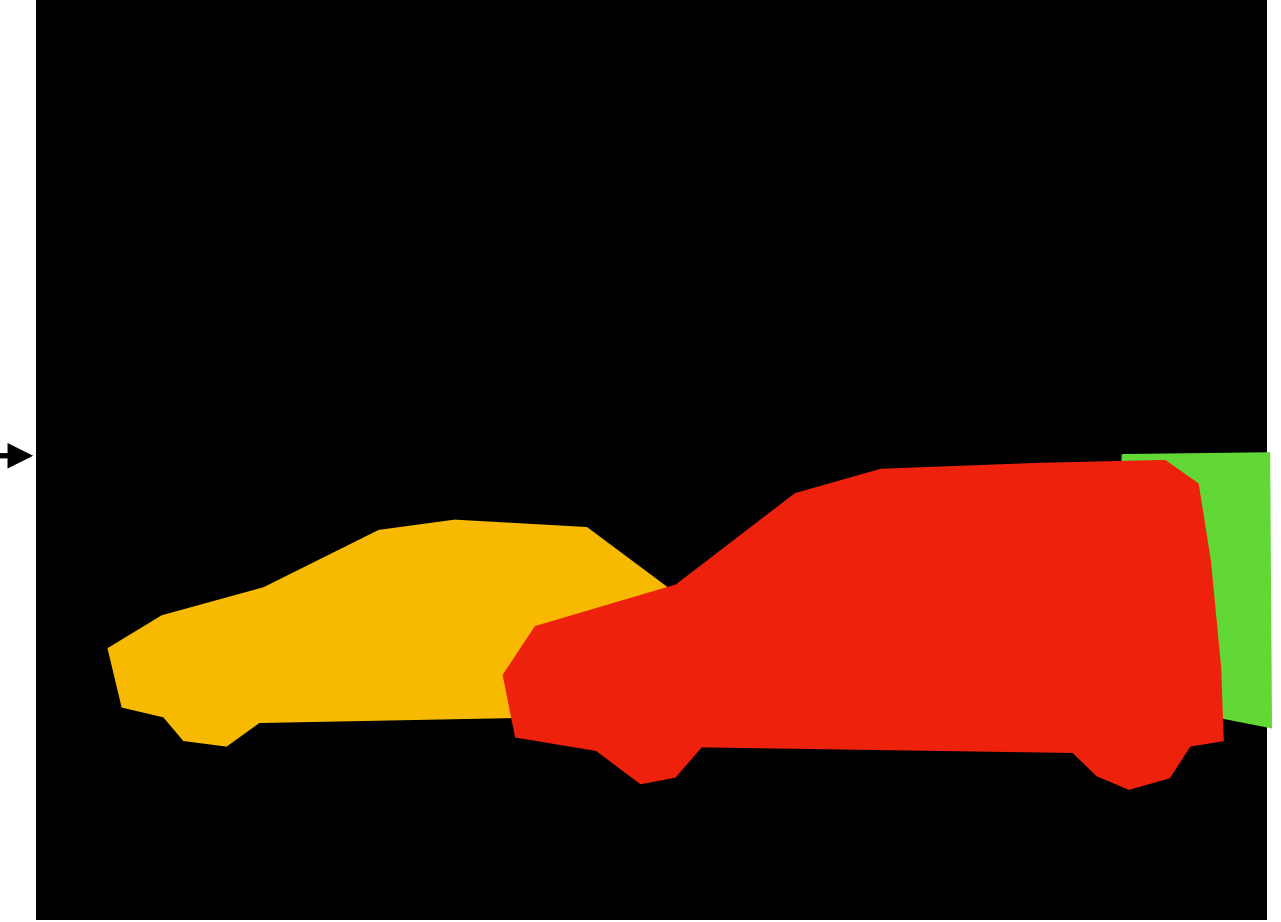
Is instance i of object class c present in the pixel $[n,m]$?



Segmentation



Instance-segmentation

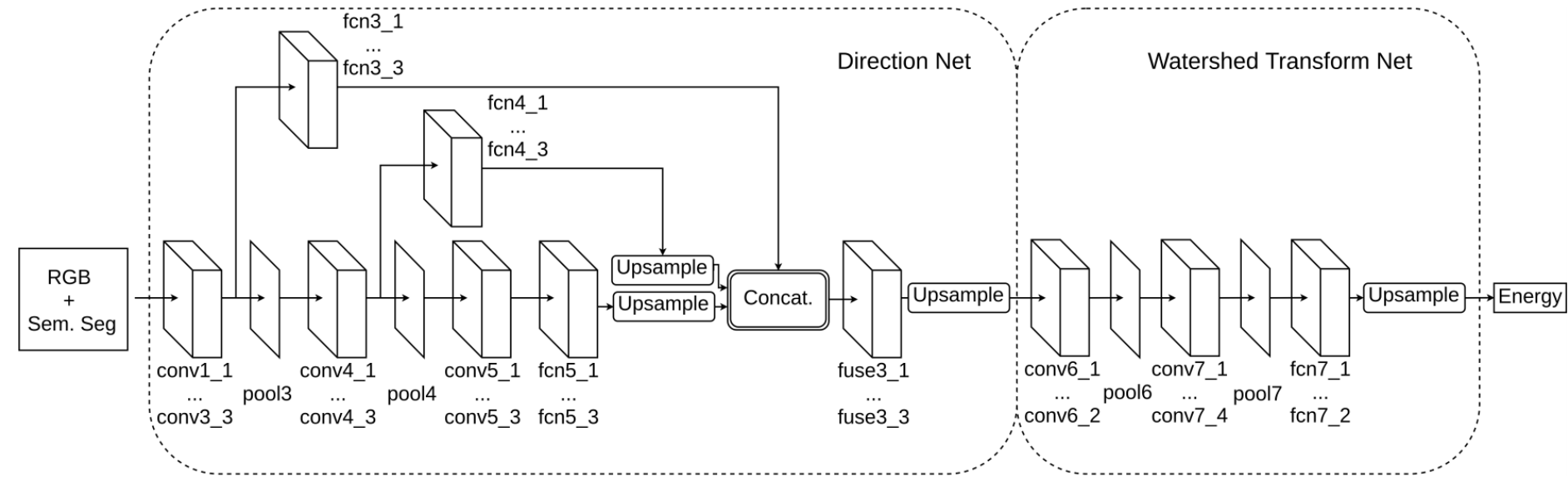


Instance segmentation

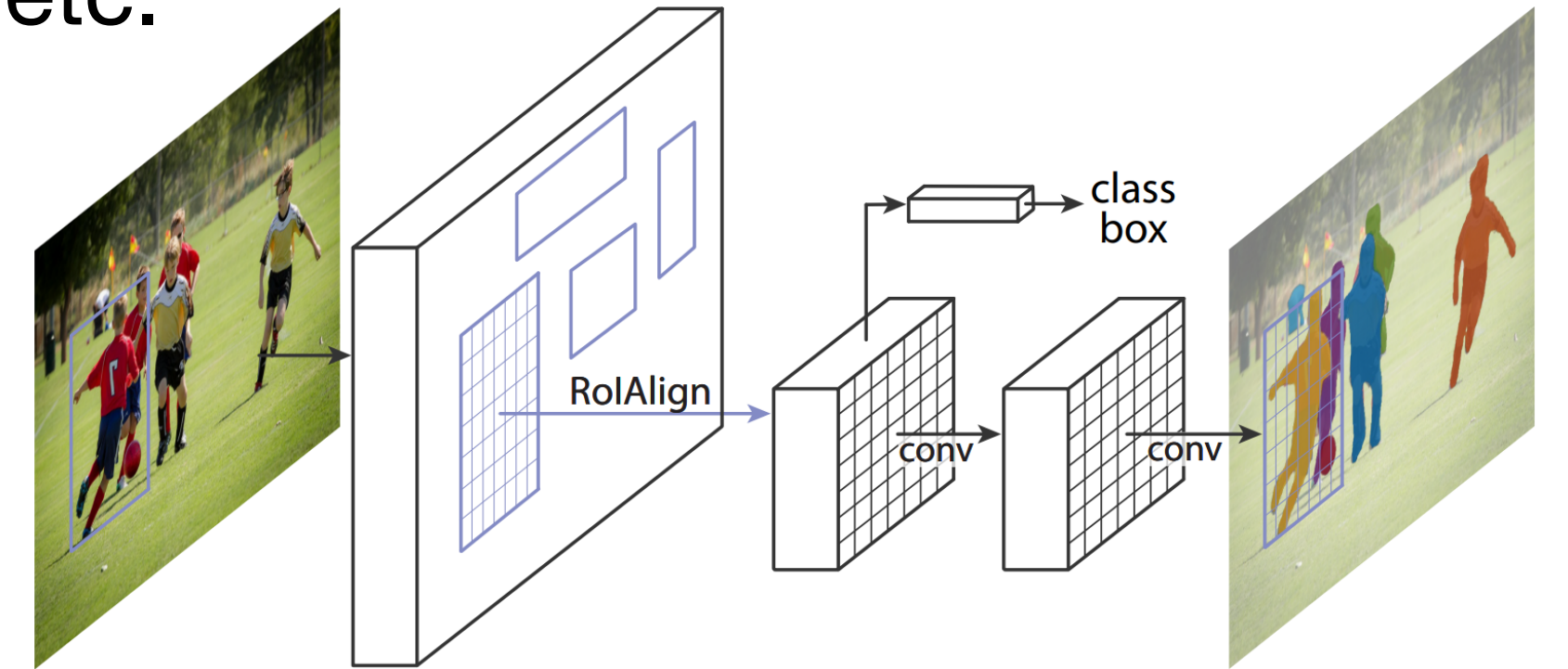


Approaches

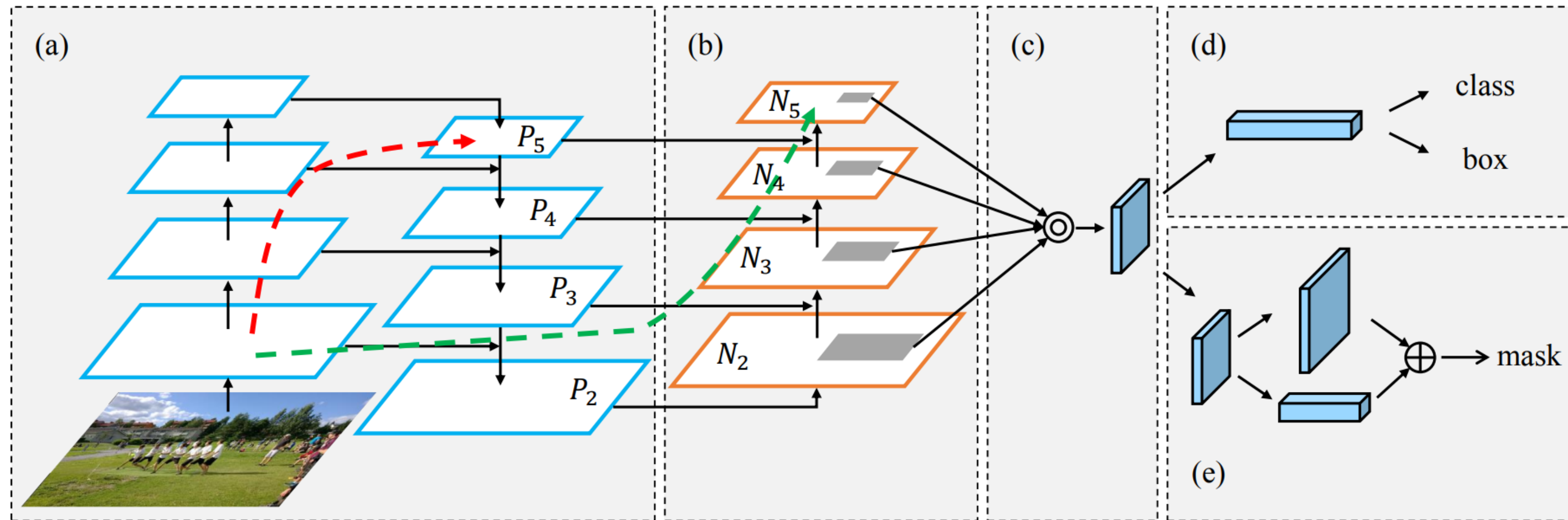
InstanceCut, DWT, SAIS, DIN, FCIS, SGN, Mask-RCNN, PANet etc.



DWT [Bai et al, CVPR'17]



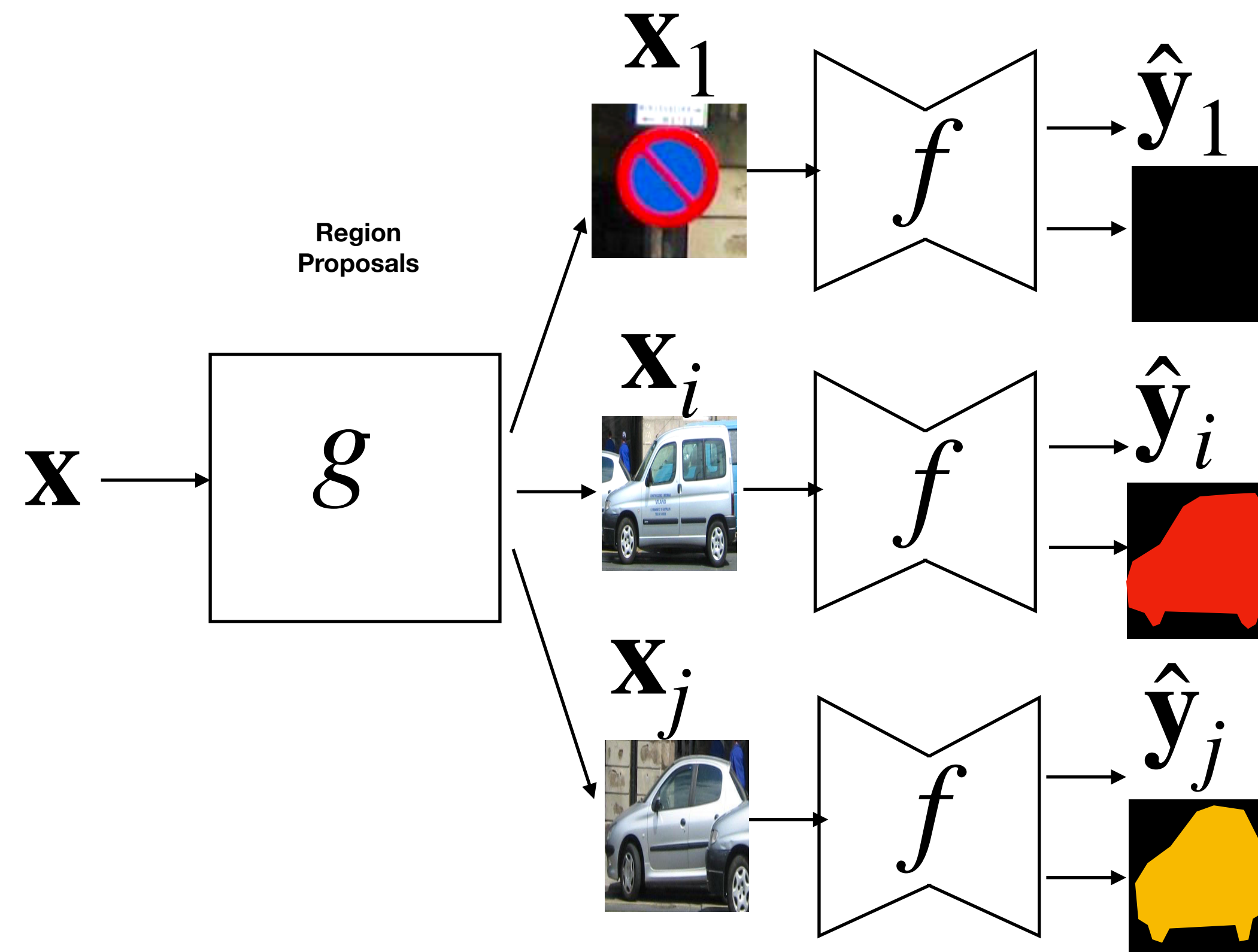
Mask-RCNN [He et al, ICCV'17]



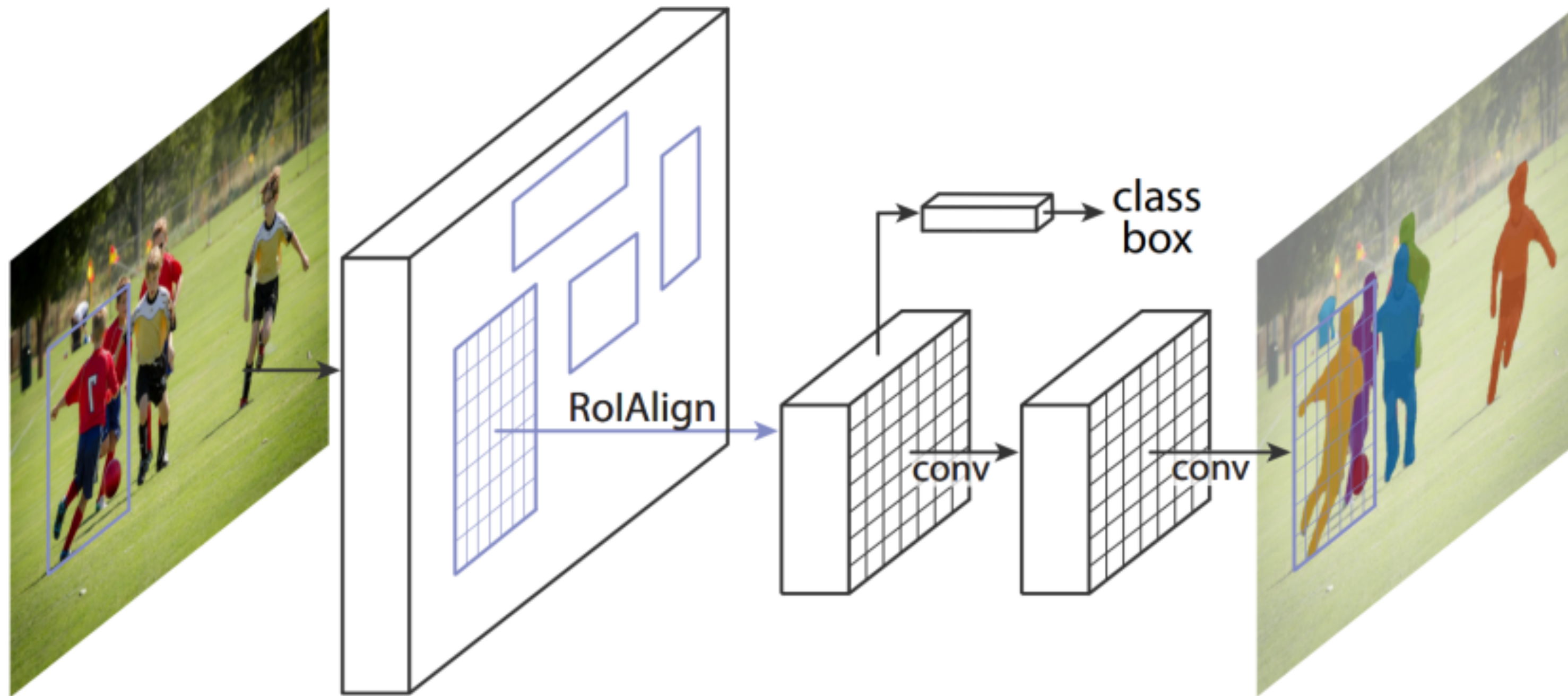
PANet [Liu et al, CVPR'18]

Solution:

Combine classification, regions proposal and segmentation



Mask R-CNN



Mask R-CNN

Kaiming He Georgia Gkioxari Piotr Dollár Ross Girshick

Facebook AI Research (FAIR)

Abstract

We present a conceptually simple, flexible, and general framework for object instance segmentation. Our approach efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance. The method, called Mask R-CNN, extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps. Moreover, Mask R-CNN is easy to generalize to other tasks, e.g., allowing us to estimate human poses in the same framework. We show top results in all three tracks of the COCO suite of challenges, including instance segmentation, bounding-box object detection, and person keypoint detection. Without bells and whistles, Mask R-CNN outperforms all existing, single-model entries on every task, including the COCO 2016 challenge winners. We hope our simple and effective approach will serve as a solid baseline and help ease future research in instance-level recognition. Code has been made available at: <https://github.com/facebookresearch/Detectron>.

1. Introduction

The vision community has rapidly improved object detection and semantic segmentation results over a short period of time. In large part, these advances have been driven by powerful baseline systems, such as the Fast/Faster R-CNN [12, 36] and Fully Convolutional Network (FCN) [30] frameworks for object detection and semantic segmentation, respectively. These methods are conceptually intuitive and offer flexibility and robustness, together with fast training and inference time. Our goal in this work is to develop a comparably enabling framework for *instance segmentation*.

Instance segmentation is challenging because it requires the correct detection of all objects in an image while also precisely segmenting each instance. It therefore combines elements from the classical computer vision tasks of *object detection*, where the goal is to classify individual objects and localize each using a bounding box, and *semantic*

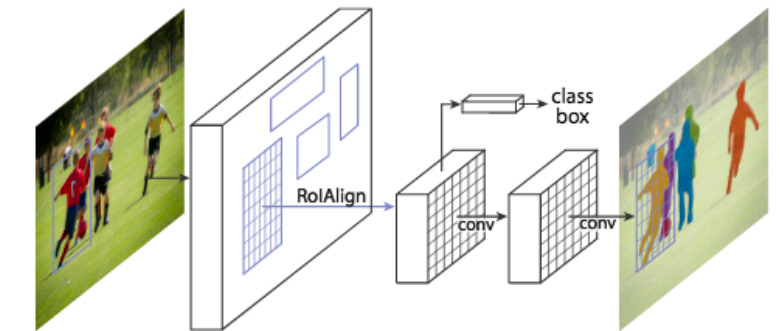


Figure 1. The Mask R-CNN framework for instance segmentation.

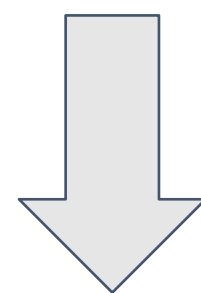
segmentation, where the goal is to classify each pixel into a fixed set of categories without differentiating object instances.¹ Given this, one might expect a complex method is required to achieve good results. However, we show that a surprisingly simple, flexible, and fast system can surpass prior state-of-the-art instance segmentation results.

Our method, called *Mask R-CNN*, extends Faster R-CNN [36] by adding a branch for predicting segmentation masks on each Region of Interest (RoI), in *parallel* with the existing branch for classification and bounding box regression (Figure 1). The mask branch is a small FCN applied to each RoI, predicting a segmentation mask in a pixel-to-pixel manner. Mask R-CNN is simple to implement and train given the Faster R-CNN framework, which facilitates a wide range of flexible architecture designs. Additionally, the mask branch only adds a small computational overhead, enabling a fast system and rapid experimentation.

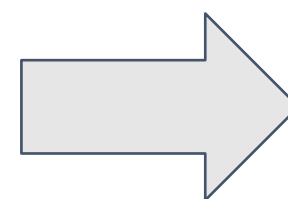
In principle Mask R-CNN is an intuitive extension of Faster R-CNN, yet constructing the mask branch properly is critical for good results. Most importantly, Faster R-CNN was not designed for pixel-to-pixel alignment between network inputs and outputs. This is most evident in how *RoIPool* [18, 12], the *de facto* core operation for attending to instances, performs coarse spatial quantization for feature extraction. To fix the misalignment, we propose a simple, quantization-free layer, called *RoIAlign*, that faithfully preserves exact spatial locations. Despite being

¹Following common terminology, we use *object detection* to denote detection via *bounding boxes*, not masks, and *semantic segmentation* to denote per-pixel classification without differentiating instances. Yet we note that *instance segmentation* is both semantic and a form of detection.

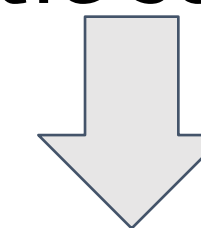
Panoptic Segmentation



Instance detection



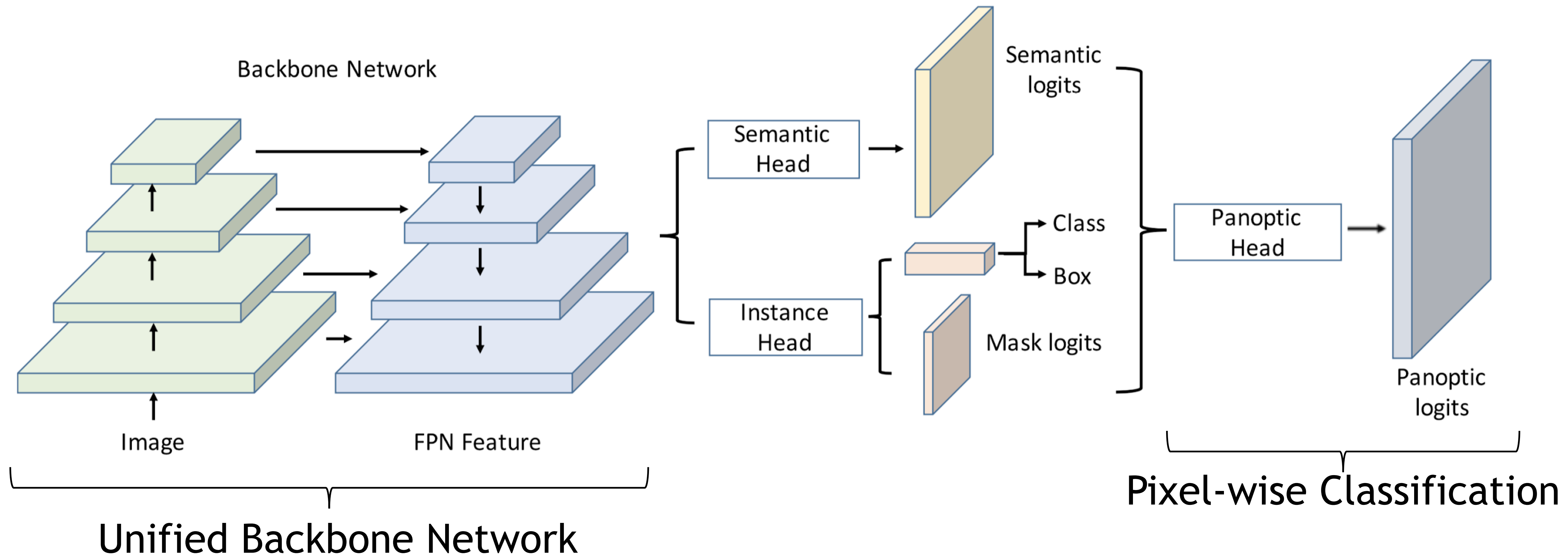
Semantic segmentation



panoptic segmentation:

stuff and things are solved, instances distinguishable


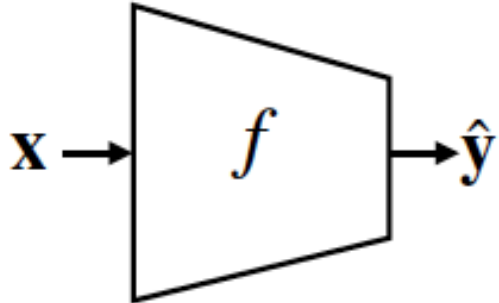
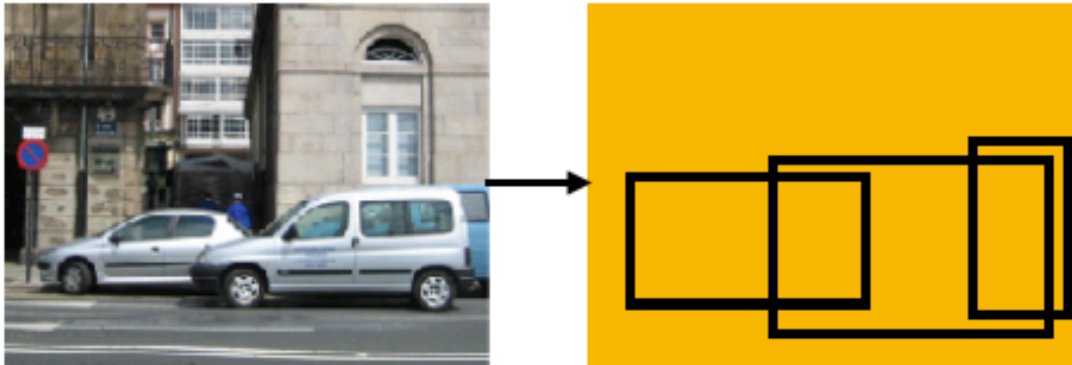
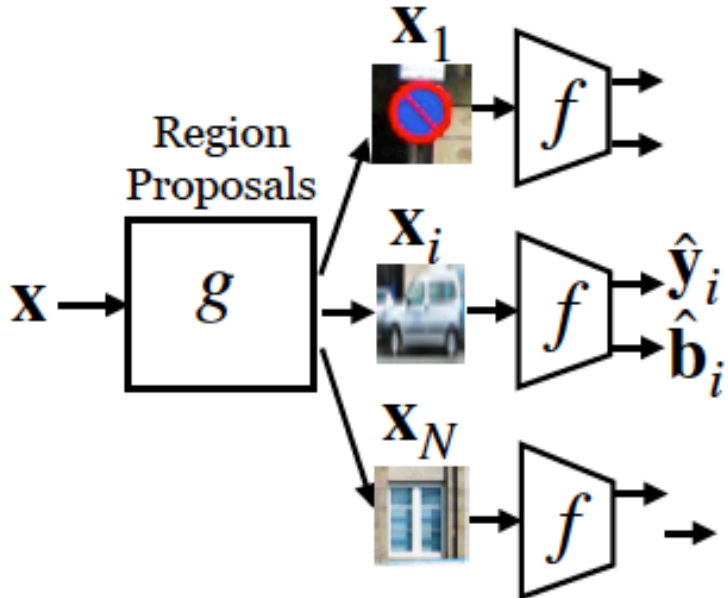
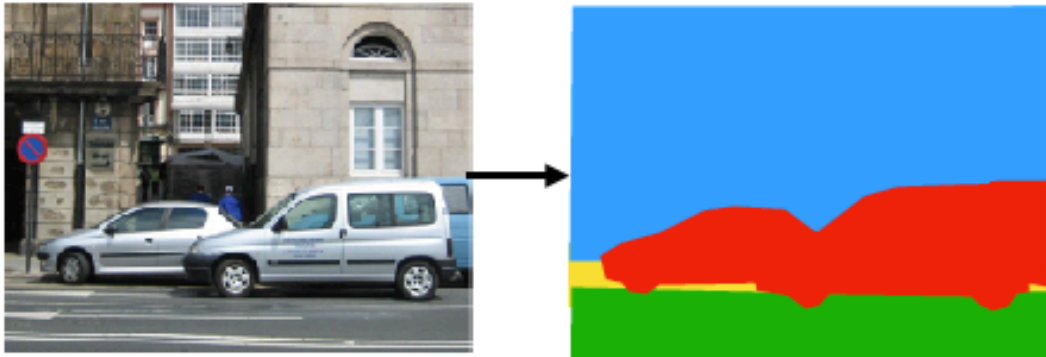
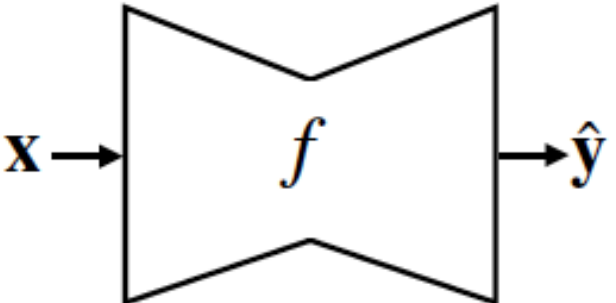
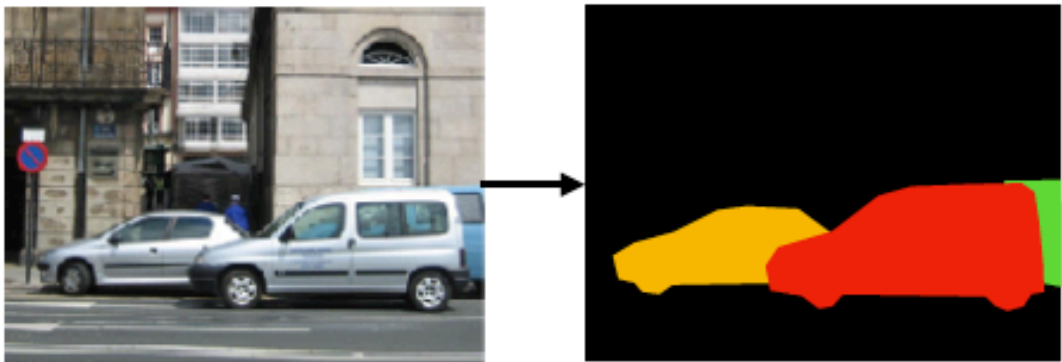
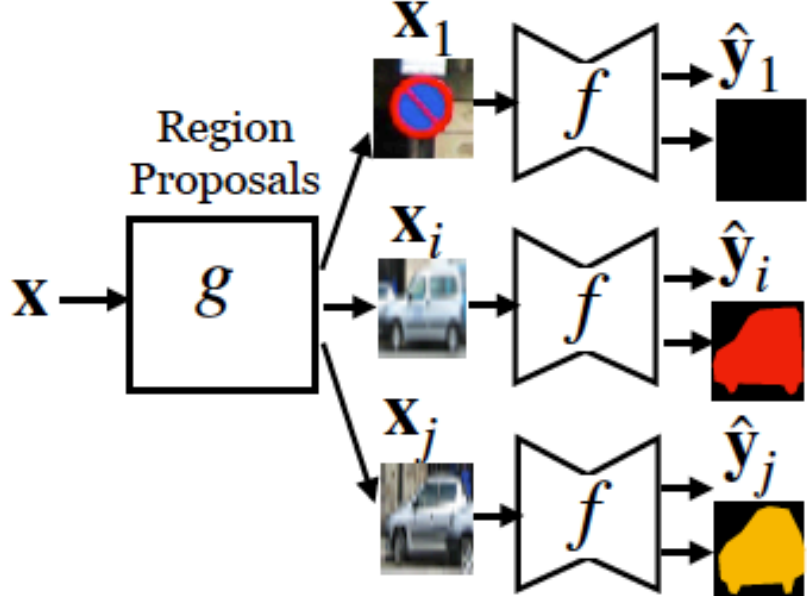
Unified Panoptic Segmentation Network (UPSNNet)



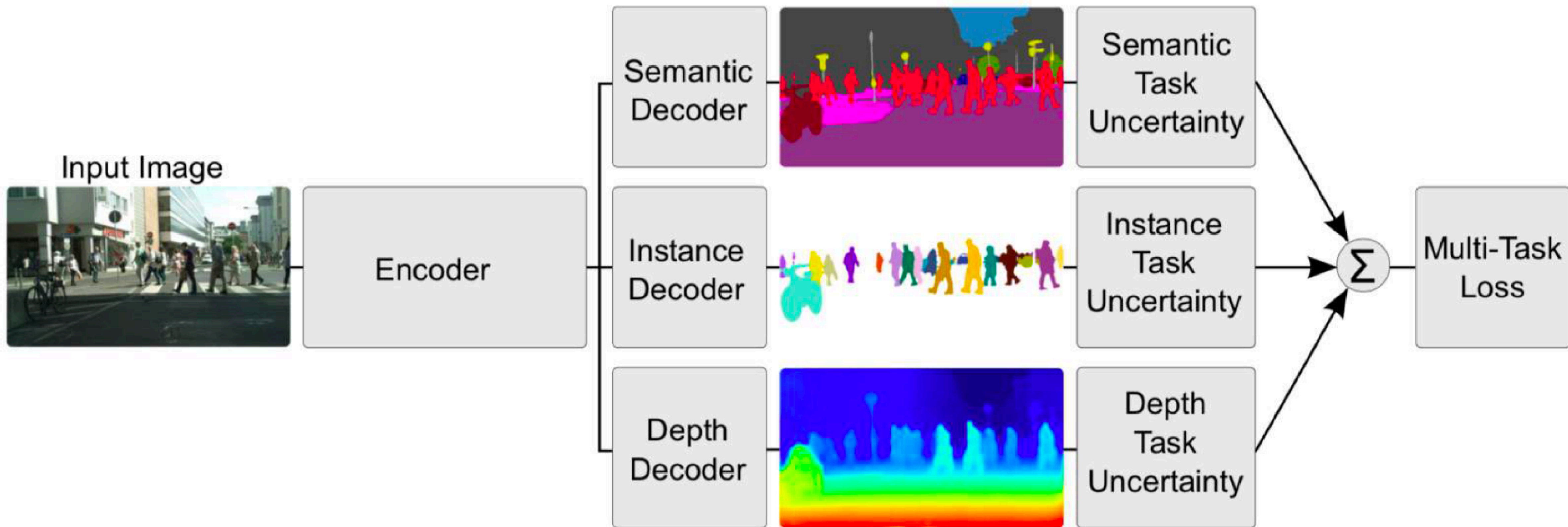
UPSNet-101-M: Cityscapes



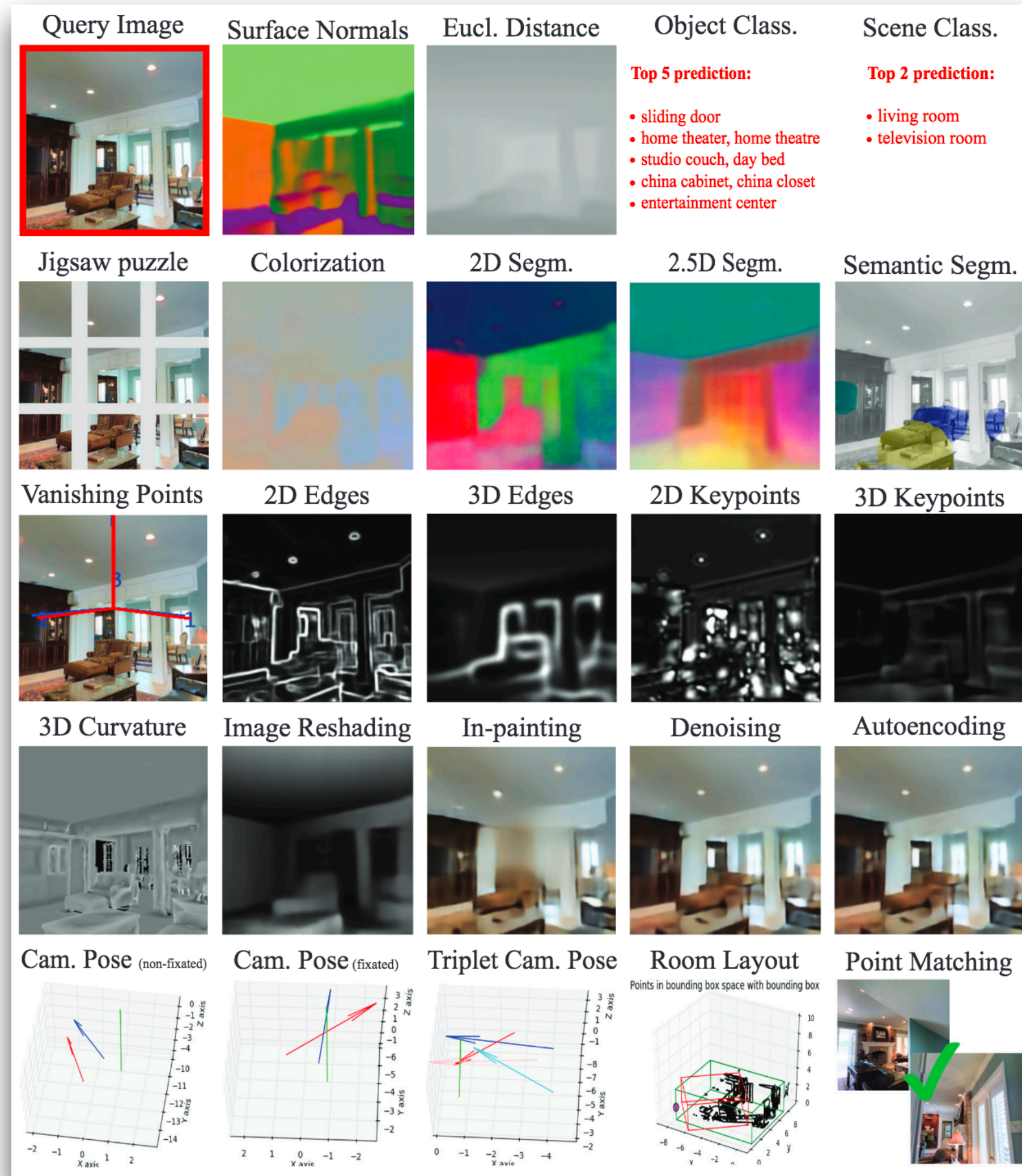
Summary

	Question	Mapping	Architecture	Functional form
Classification	Is object class c present in the image?	 <div>Car Building Road</div>		$\hat{y} = f(\mathbf{x})$
Detection	What is the location of all the instances of class c ?			$\{\mathbf{x}_i\}_{i=1}^N = g(\mathbf{x})$ $(\hat{y}_i, \hat{\mathbf{b}}_i) = f(\mathbf{x}_i)$
Segmentation	Is object class c present in the pixel $[n,m]$?			$\hat{y}[n, m] = f(\mathbf{x})$
Instance segmentation	Is instance i of object class c present in the pixel $[n,m]$?			$\{\mathbf{x}_i\}_{i=1}^N = g(\mathbf{x})$ $(\hat{y}_i, \hat{s}_i[n, m]) = f(\mathbf{x}_i)$

Scene understanding



Taskonomy



arXiv:1804.08328v1 [cs.CV] 23 Apr 2018

Taskonomy: Disentangling Task Transfer Learning

Amir R. Zamir^{1,2} Alexander Sax^{1*} William Shen^{1*} Leonidas Guibas¹ Jitendra Malik² Silvio Savarese¹

¹ Stanford University ² University of California, Berkeley

<http://taskonomy.vision/>

Abstract

Do visual tasks have a relationship, or are they unrelated? For instance, could having surface normals simplify estimating the depth of an image? Intuition answers these questions positively, implying existence of a **structure** among visual tasks. Knowing this structure has notable values; it is the concept underlying transfer learning and provides a principled way for identifying redundancies across tasks, e.g., to seamlessly reuse supervision among related tasks or solve many tasks in one system without piling up the complexity.

We propose a fully computational approach for modeling the structure of space of visual tasks. This is done via finding (first and higher-order) transfer learning dependencies across a dictionary of twenty six 2D, 2.5D, 3D, and semantic tasks in a latent space. The product is a computational taxonomic map for task transfer learning. We study the consequences of this structure, e.g. nontrivial emerged relationships, and exploit them to reduce the demand for labeled data. For example, we show that the total number of labeled datapoints needed for solving a set of 10 tasks can be reduced by roughly $\frac{2}{3}$ (compared to training independently) while keeping the performance nearly the same. We provide a set of tools for computing and probing this taxonomical structure including a solver that users can employ to devise efficient supervision policies for their use cases.

1. Introduction

Object recognition, depth estimation, edge detection, pose estimation, etc are examples of common vision tasks deemed useful and tackled by the research community. Some of them have rather clear relationships: we understand that surface normals and depth are related (one is a derivate of the other), or vanishing points in a room are useful for orientation. Other relationships are less clear: how keypoint detection and the shading in a room can, together, perform pose estimation.

*Equal.

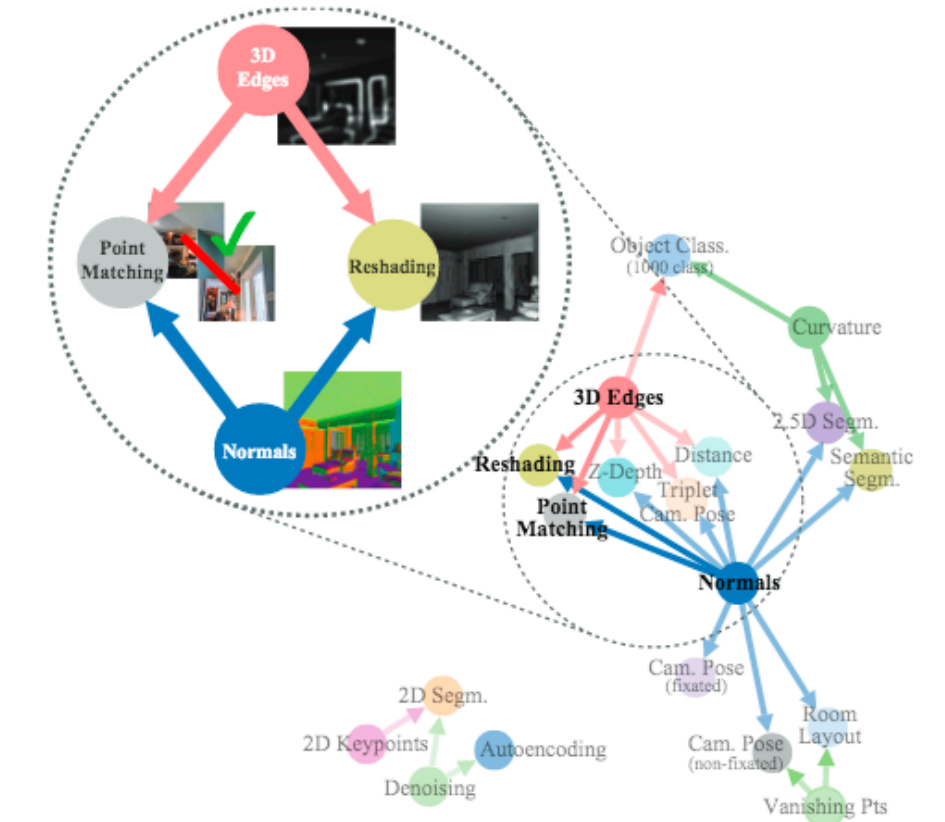


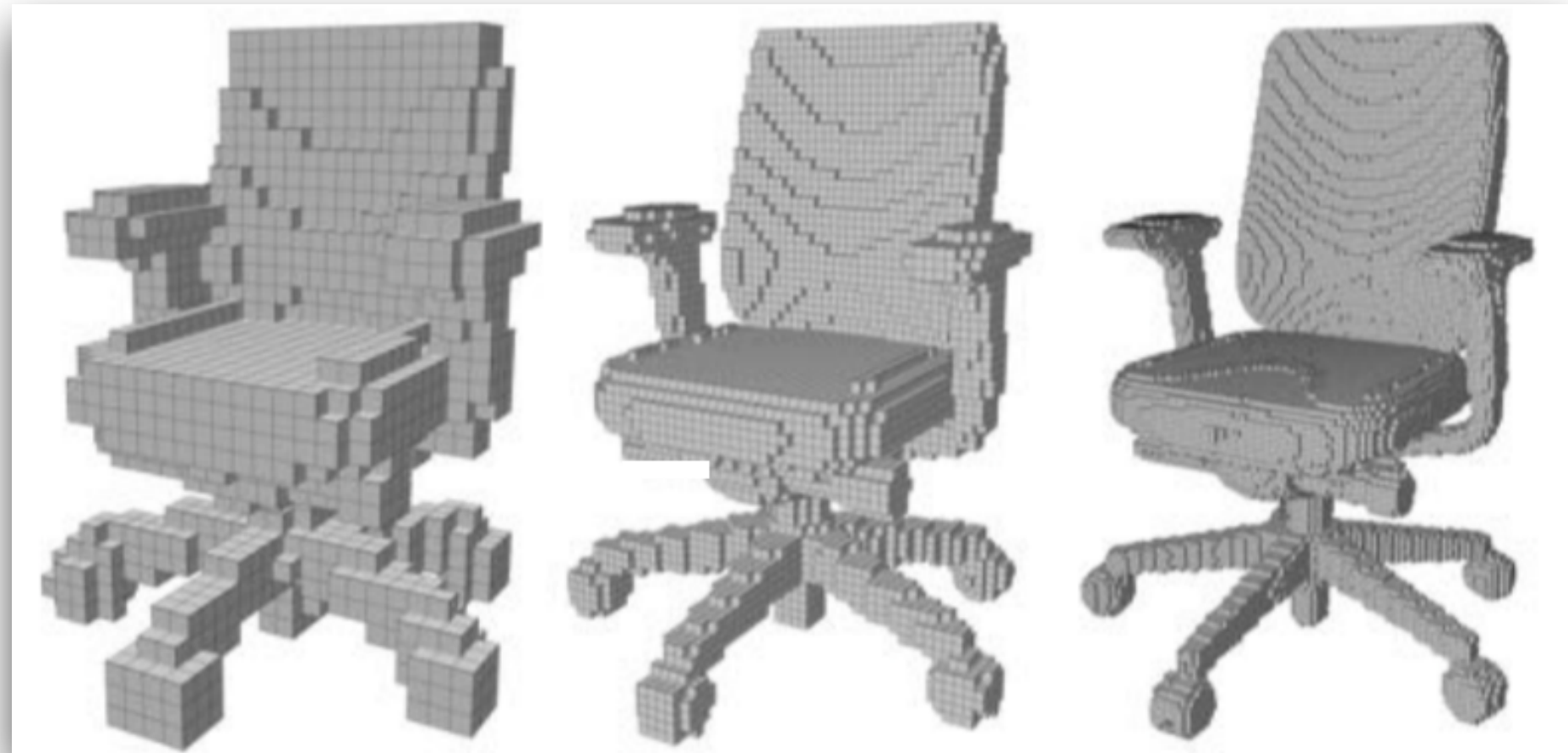
Figure 1: A sample task structure discovered by the computational task taxonomy (*taskonomy*). It found that, for instance, by combining the learned features of a surface normal estimator and occlusion edge detector, good networks for reshading and point matching can be rapidly trained with little labeled data.

The field of computer vision has indeed gone far without explicitly using these relationships. We have made remarkable progress by developing advanced learning machinery (e.g. ConvNets) capable of finding complex mappings from X to Y when many pairs of (x, y) s.t. $x \in X, y \in Y$ are given as training data. This is usually referred to as fully supervised learning and often leads to problems being solved in isolation. Siloing tasks makes training a new task or a comprehensive perception system a Sisyphean challenge, whereby each task needs to be learned individually from scratch. Doing so ignores their quantifiably useful relationships leading to a massive labeled data requirement.

Alternatively, a model aware of the relationships among tasks demands less supervision, uses less computation, and behaves in more predictable ways. Incorporating such a structure is the first stepping stone towards develop-

3D Object Models

Voxels



3D Keypoint



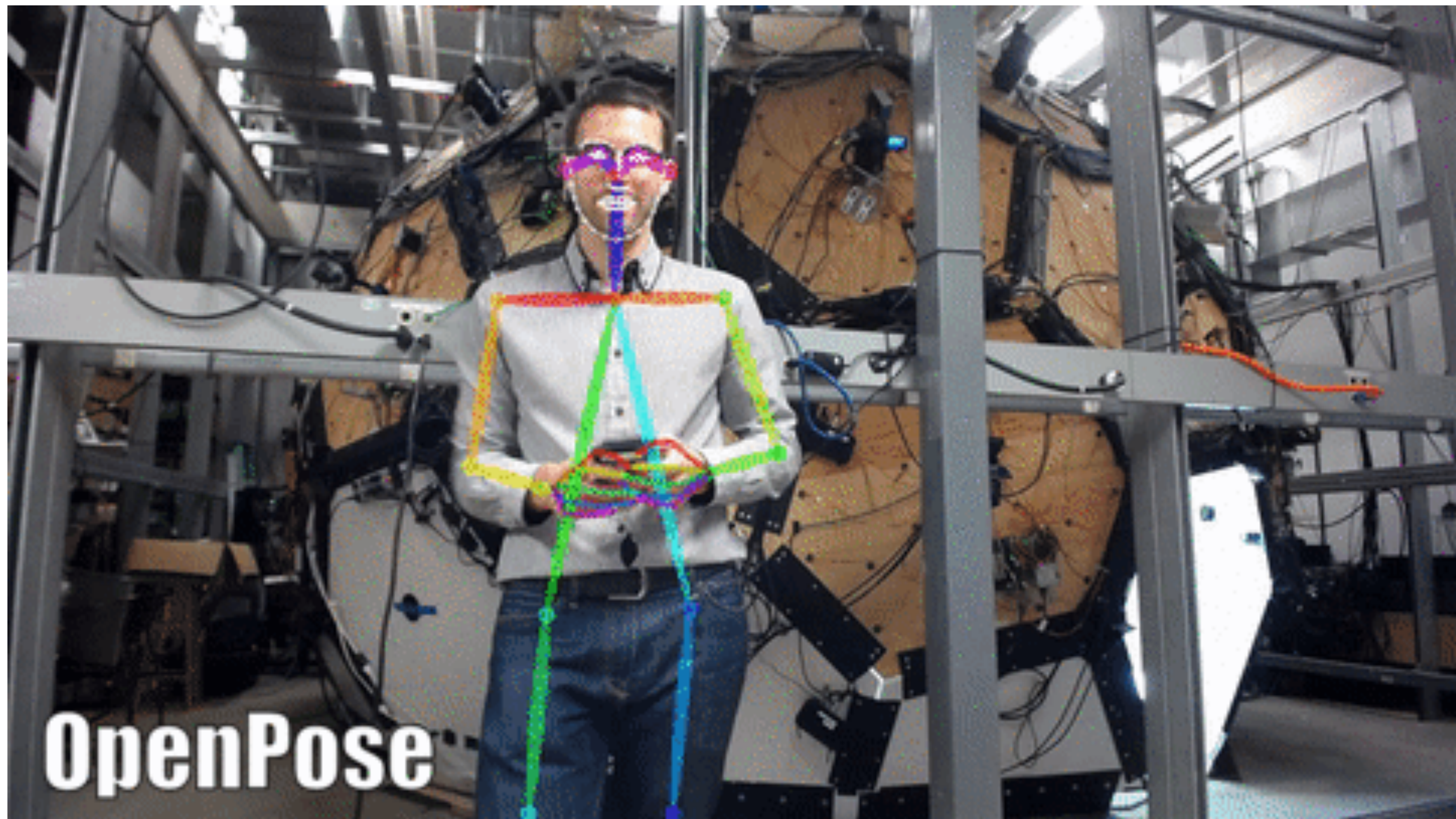
Multiviews



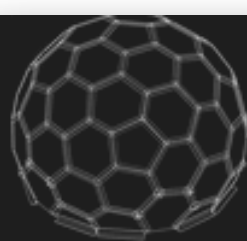
Meshes



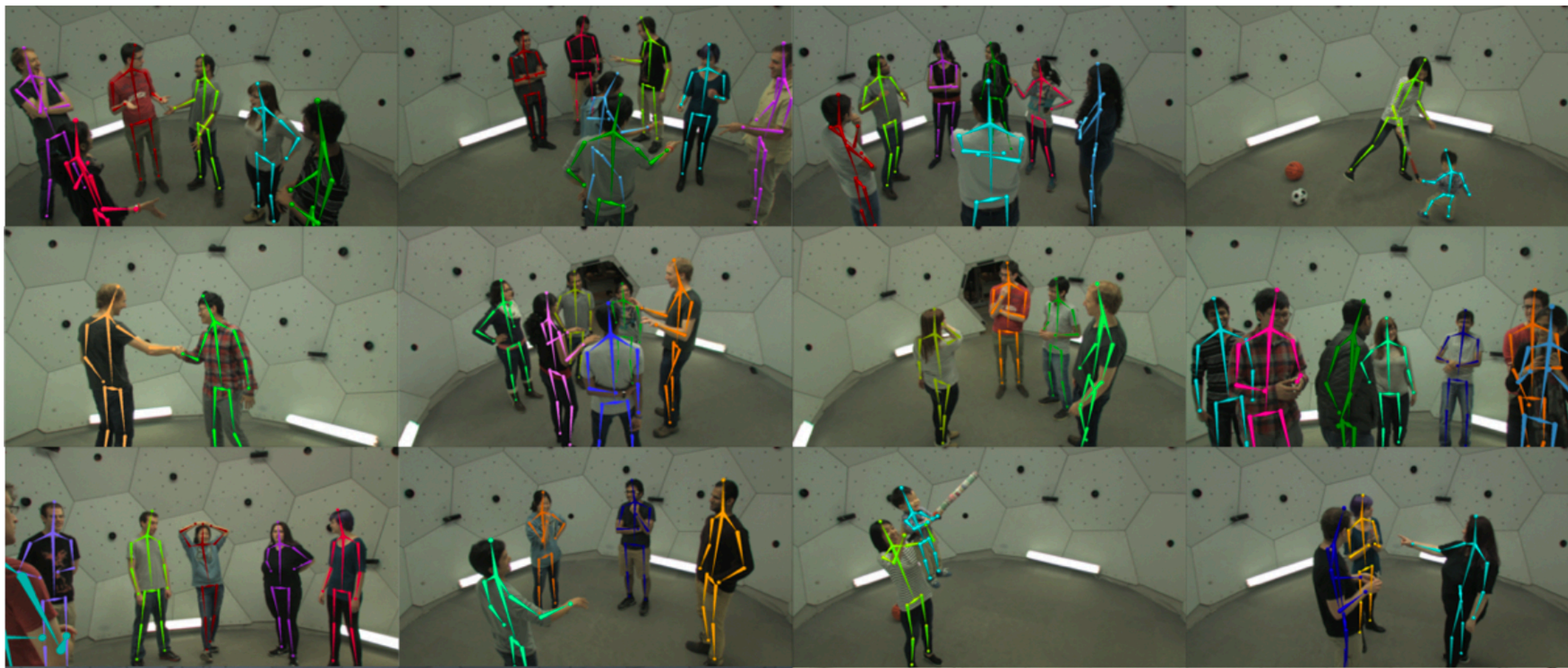
People



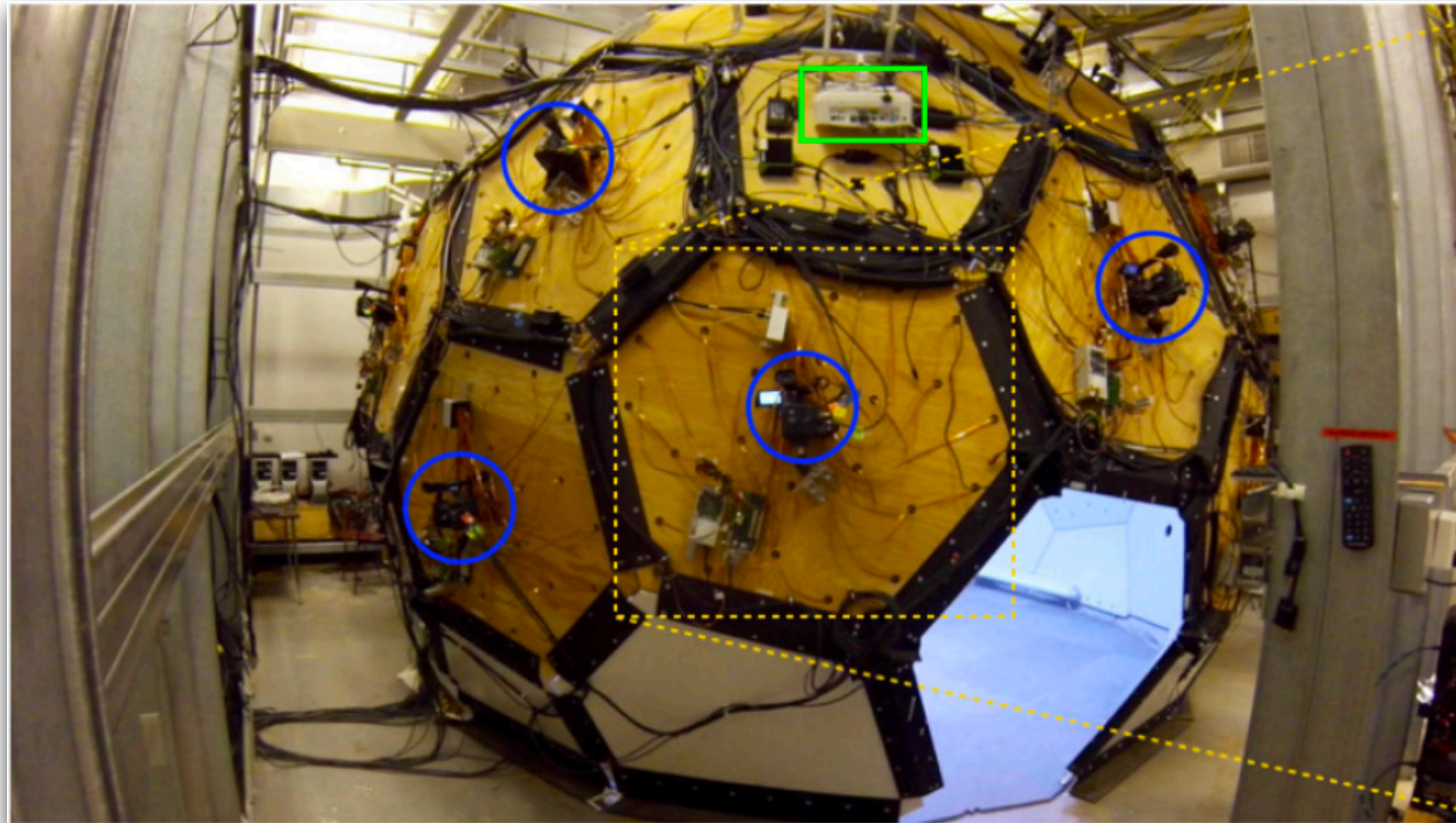
https://github.com/CMU-Perceptual-Computing-Lab/openpose/blob/master/doc/media/pose_face_hands.gif



Dataset Examples



Panoptic studio



VGA cameras

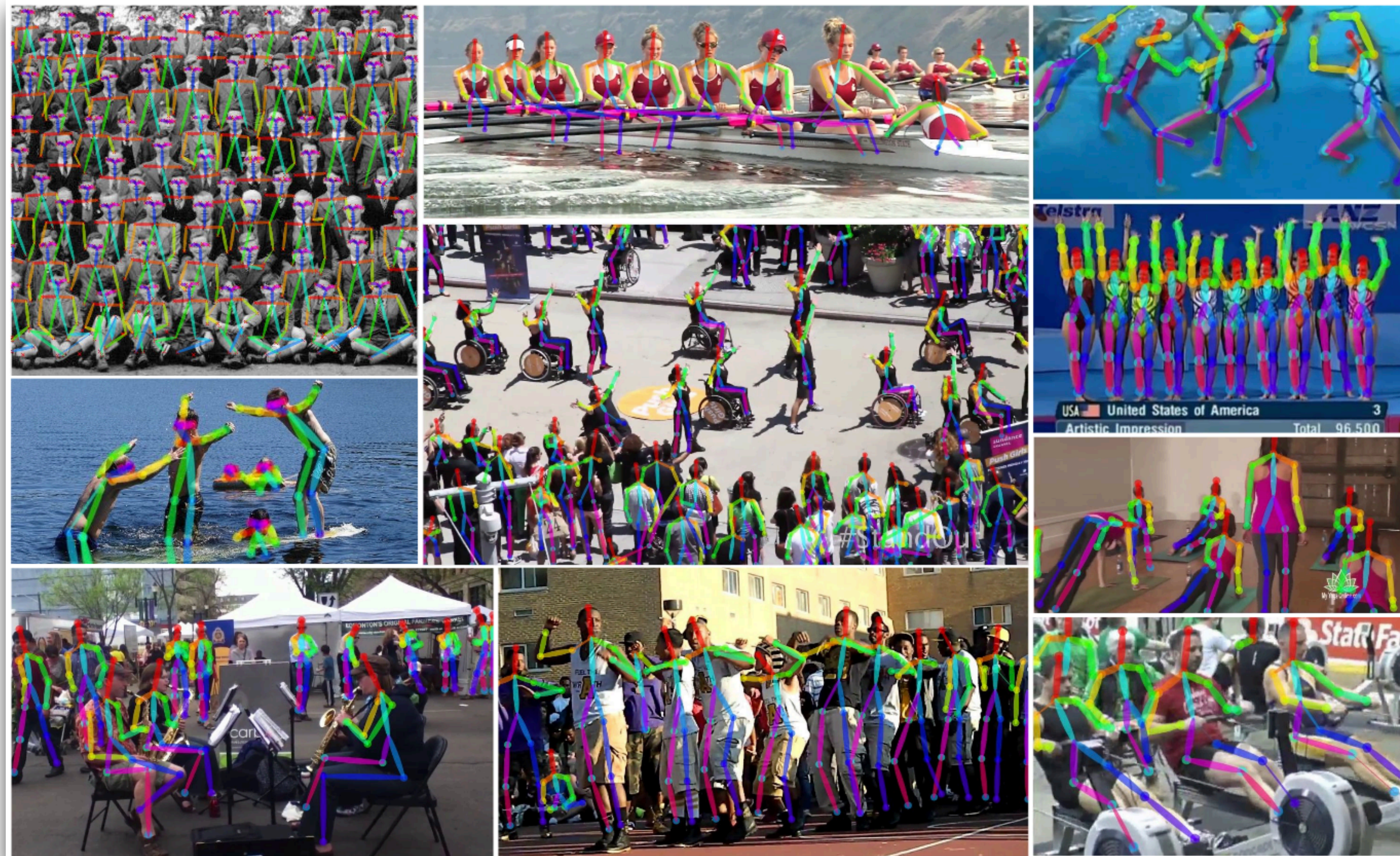
HD cameras

Kinects

SFM for calibration



People



Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields *

Zhe Cao Tomas Simon Shih-En Wei Yaser Sheikh
The Robotics Institute, Carnegie Mellon University
{zhcao, shihenw}@cmu.edu {tsimon, yaser}@cs.cmu.edu

Abstract

An approach to efficiently detect the 2D pose of people in an image. The approach uses a non-representation, which we refer to as Part Affinity Fields to learn to associate body parts with individuals. The architecture encodes global context via a greedy bottom-up parsing step that maintains accuracy while achieving realtime performance, independent of the number of people in the image. The architecture is designed to jointly learn part locations and person locations via two branches of the same sequential process. Our method placed first in the inaugural 2016 keypoints challenge, and significantly exceeds state-of-the-art result on the MPII Multi-Pose dataset, both in performance and efficiency.

Introduction

Multi-person pose estimation—the problem of localizing the pose of multiple people in images, especially engaged individuals, presents a unique set of challenges. First, each image may contain an unknown number of people that can occur at any position or scale. Second, interactions between people induce complex spatial relationships, due to contact, occlusion, and limb articulation, making association of parts difficult. Third, runtime complexity grows with the number of people in the image, making it a challenge for realtime performance.

Previous approaches [23, 9, 27, 12, 19] are to employ a top-down approach and perform single-person pose estimation. These top-down approaches disintegrate existing techniques for single-person pose estimation, but suffer from commitment: if the person detector fails—as it often does when people are in close proximity—there is no recovery. Furthermore, the runtime of these

<https://youtu.be/pW6n2Xew1GM>



Figure 1. **Top:** Multi-person pose estimation. Body parts belonging to the same person are linked. **Bottom left:** Part Affinity Fields (PAFs) corresponding to the limb connecting right elbow and right wrist. The color encodes orientation. **Bottom right:** A zoomed in view of the predicted PAFs. At each pixel in the field, a 2D vector encodes the position and orientation of the limbs.

top-down approaches is proportional to the number of people: for each detection, a single-person pose estimator is run, and the more people there are, the greater the computational cost. In contrast, bottom-up approaches are attractive as they offer robustness to early commitment and have the potential to decouple runtime complexity from the number of people in the image. Yet, bottom-up approaches do not directly use global contextual cues from other body parts and other people. In practice, previous bottom-up methods [22, 11] do not retain the gains in efficiency as the final parse requires costly global inference. For example, the seminal work of Pishchulin et al. [22] proposed a bottom-up approach that jointly labeled part detection candidates and associated them to individual people. However, solving the integer linear programming problem over a fully connected graph is an NP-hard problem and the average processing time is on the order of hours. Insafutdinov et al. [11] built on [22] with stronger part detectors based on ResNet [10] and image-dependent pairwise scores, and vastly improved the runtime, but the method still takes several minutes per image, with a limit on the number of part proposals. The pairwise representations used in [11], are difficult to regress precisely and thus a separate logistic regression is required.

Stacked hourglass architecture



arXiv:1603.06937v2 [cs.CV] 26 Jul 2016

Stacked Hourglass Networks for Human Pose Estimation

Alejandro Newell, Kaiyu Yang, and Jia Deng

University of Michigan, Ann Arbor
{alnewell, yangky, jiadeng}@umich.edu

Abstract. This work introduces a novel convolutional network architecture for the task of human pose estimation. Features are processed across all scales and consolidated to best capture the various spatial relationships associated with the body. We show how repeated bottom-up, top-down processing used in conjunction with intermediate supervision is critical to improving the performance of the network. We refer to the architecture as a “stacked hourglass” network based on the successive steps of pooling and upsampling that are done to produce a final set of predictions. State-of-the-art results are achieved on the FLIC and MPII benchmarks outcompeting all recent methods.

Keywords: Human Pose Estimation

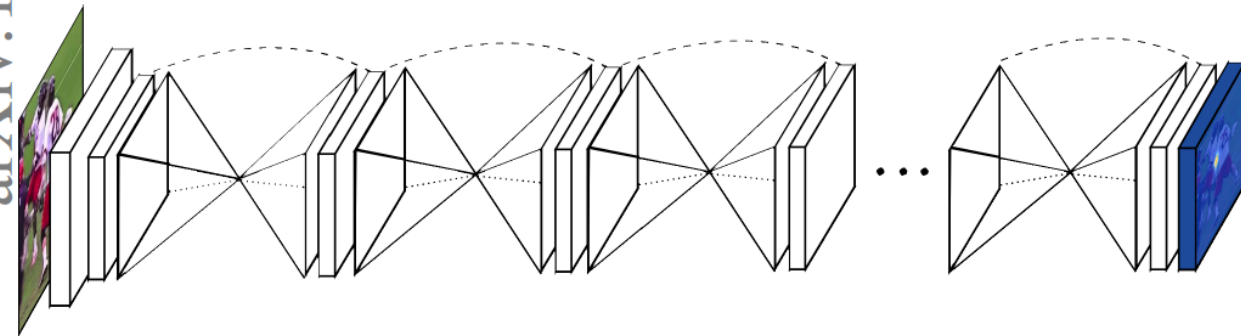
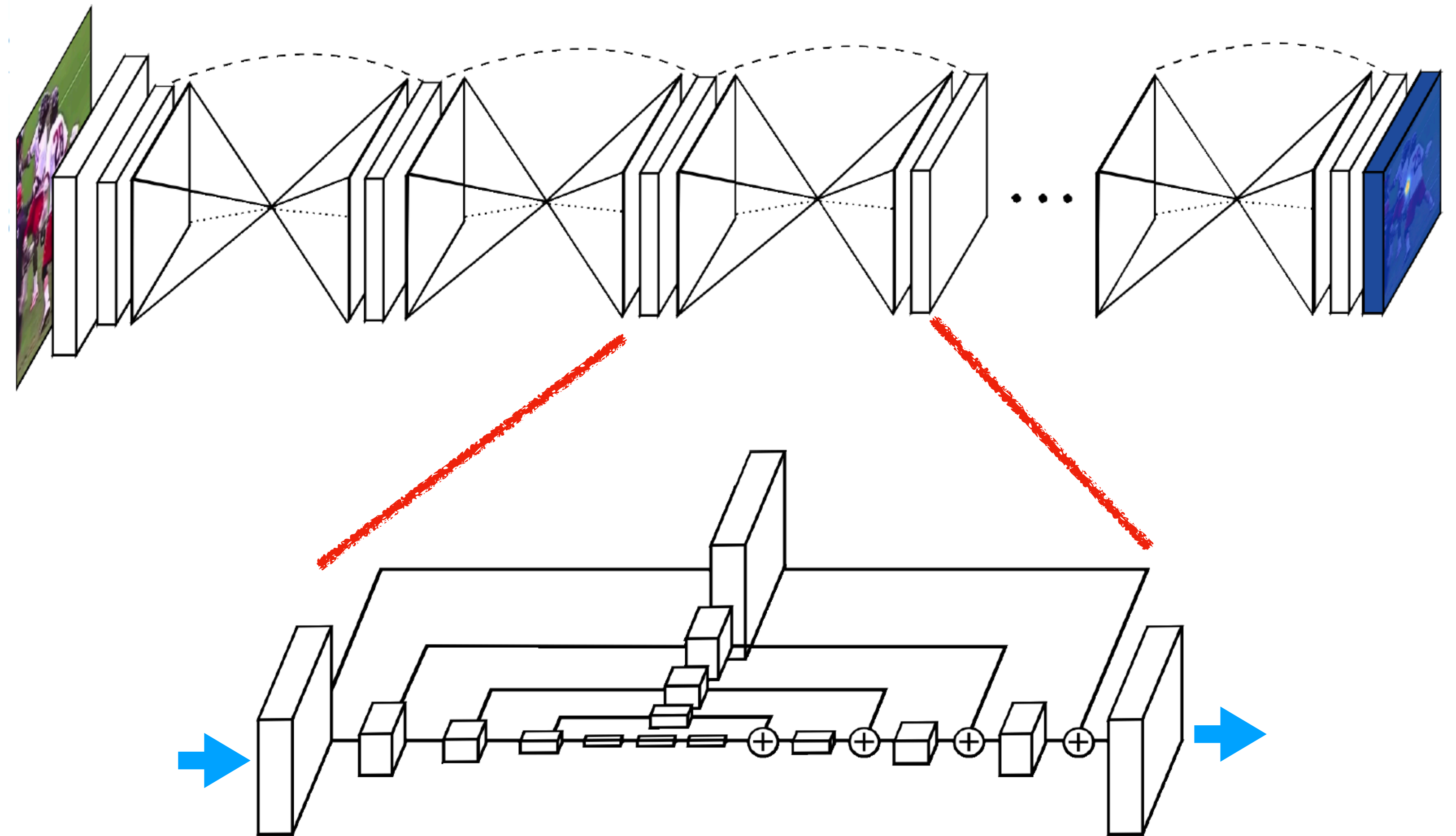


Fig. 1. Our network for pose estimation consists of multiple stacked hourglass modules which allow for repeated bottom-up, top-down inference.

1 Introduction

A key step toward understanding people in images and video is accurate pose estimation. Given a single RGB image, we wish to determine the precise pixel location of important keypoints of the body. Achieving an understanding of a person’s posture and limb articulation is useful for higher level tasks like action recognition, and also serves as a fundamental tool in fields such as human-computer interaction and animation.



Keypoint heatmaps



<https://arxiv.org/pdf/1603.06937.pdf>