Object Detection

CSC2541, 2017 Winter
Bin Yang
13 Feb. 2017

"If I have seen further it is by standing on the shoulders of giants." - Isaac Newton

slides adopted from Ross Girshick, Chris McIntosh, Sanja Fidler, Mubarak Shah and many others

Object detection

- Introduction
- Pre-CNN time
  - HOG detector
  - Deformable Part-based Model
- CNN time
  - Region-CNN
  - Fast versions of R-CNN
  - YOLO/SSD
- 3D object detection
- Devil's in the details

Object detection

- Introduction
- Pre-CNN time
    - HOG detector
    - Deformable Part-based Model
- CNN time
    - Region-CNN
    - Fast versions of R-CNN
    - YOLO/SSD
- 3D object detection
- Devil's in the details

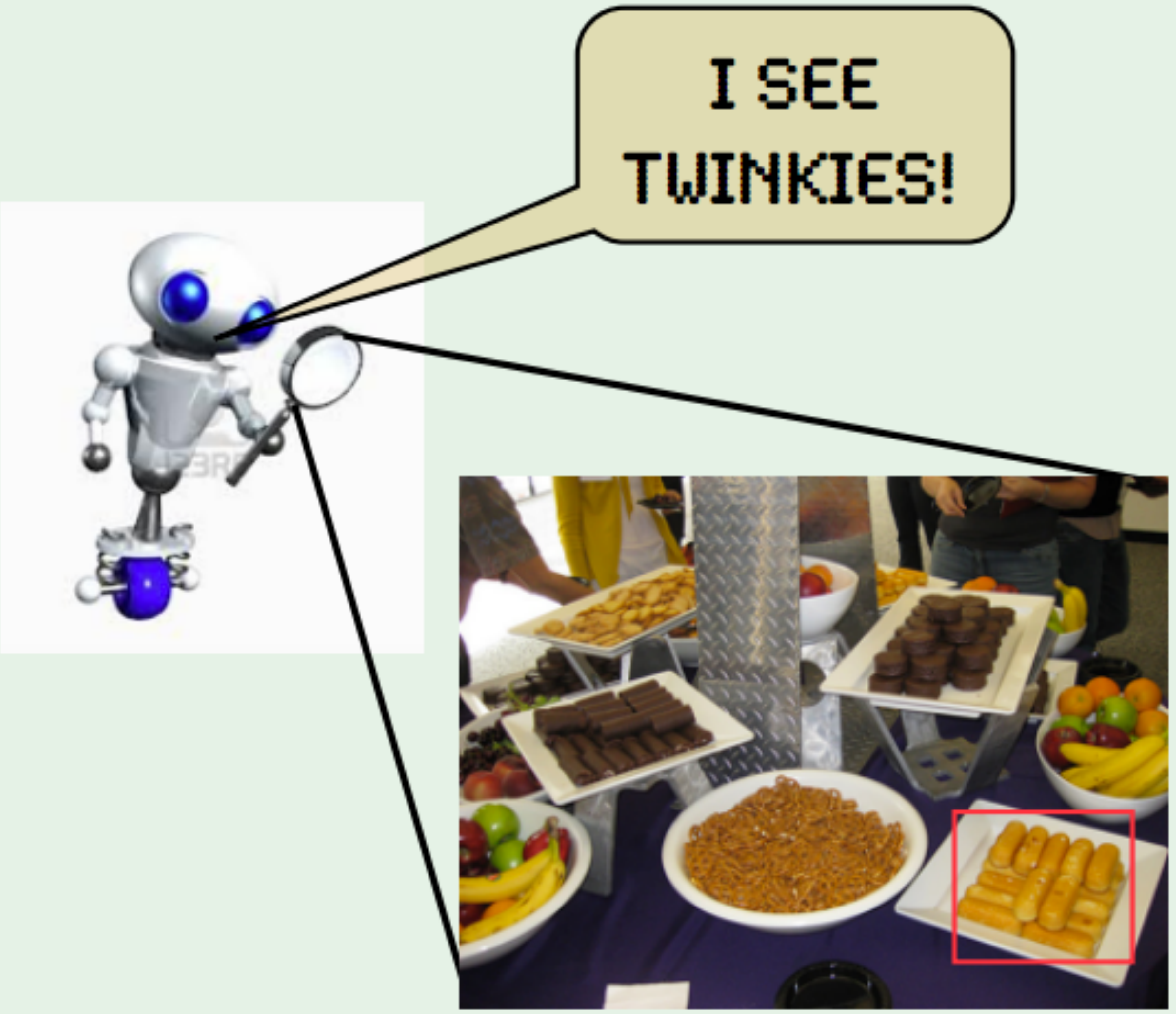
Image understanding

Snack time in the lab

photo by "thomas pix" http://www.flickr.com/photos/thomaspix/2591427106

slide credit: Ross Girshick

What objects are where?

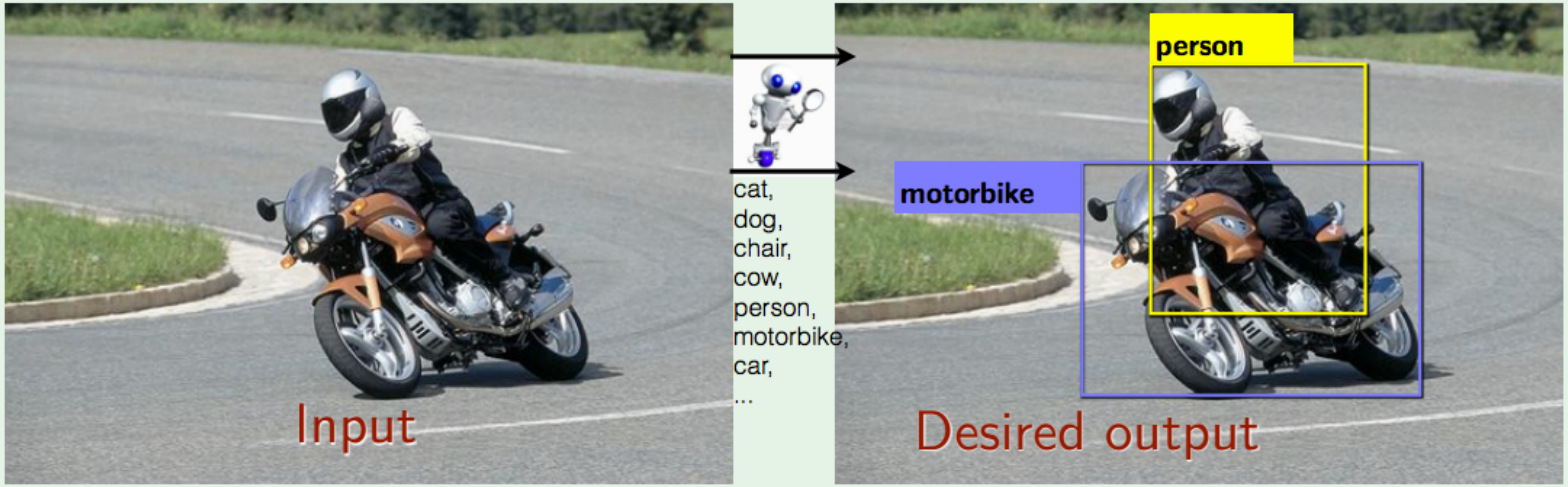


robot: “I see a table with twinkies, pretzels, fruit, and some mysterious chocolate things...”

slide credit: Ross Girshick

Formalizing the object detection task

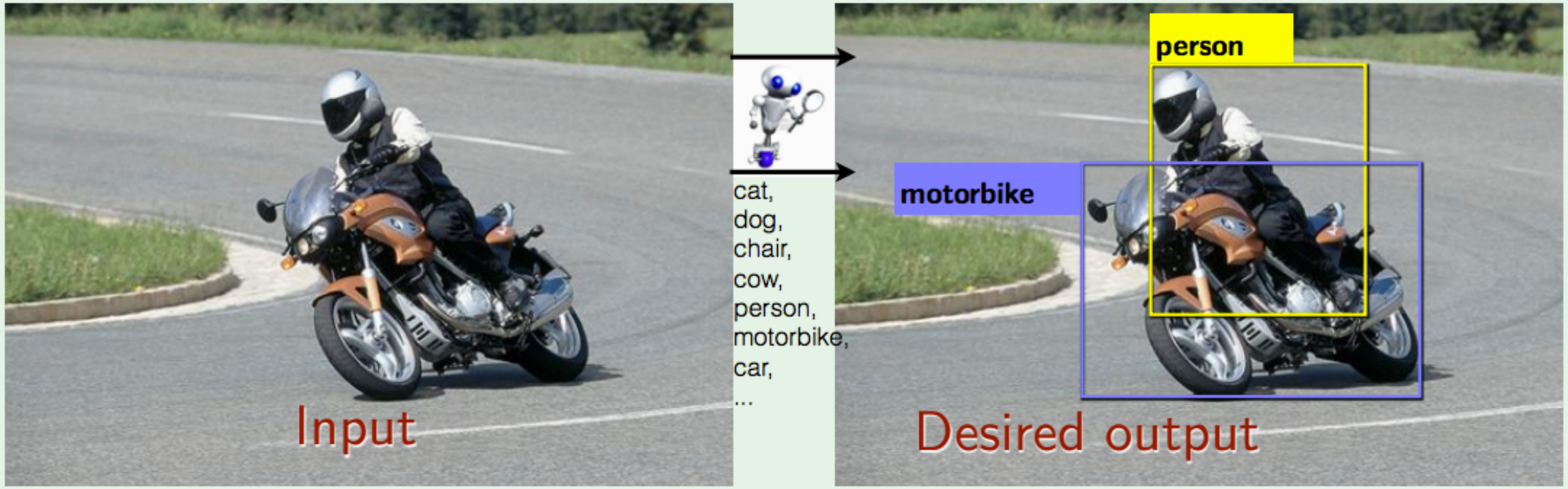
Many possible ways, this one is popular:

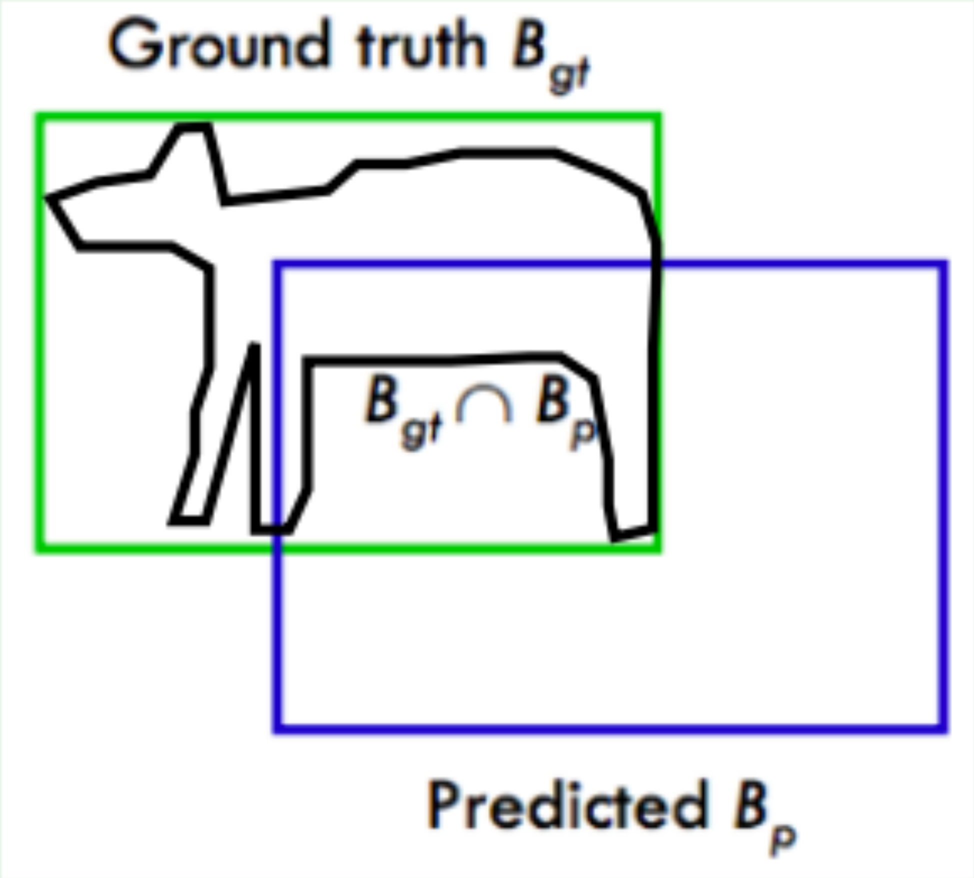


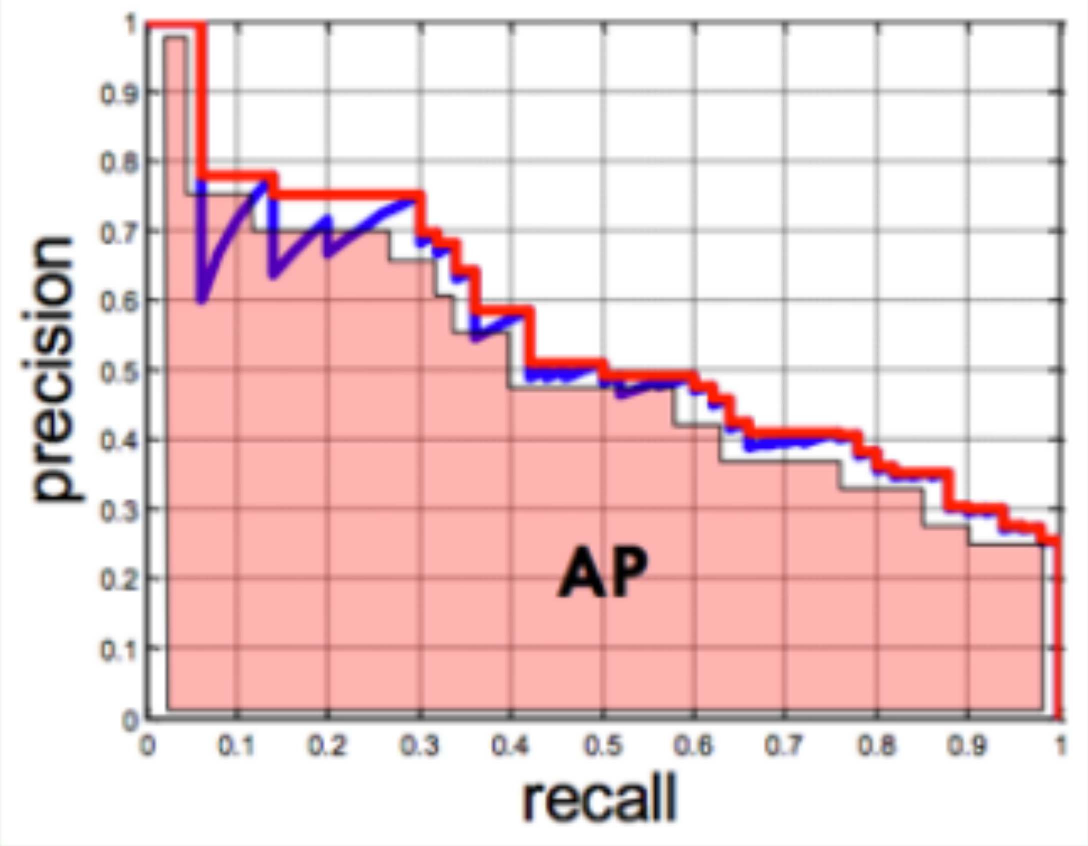
slide credit: Ross Girshick

Formalizing the object detection task

Many possible ways, this one is popular:







Performance summary:

Average Precision (AP)
0 is worst 1 is perfect

slide credit: Ross Girshick

Example 1: Find Waldo!

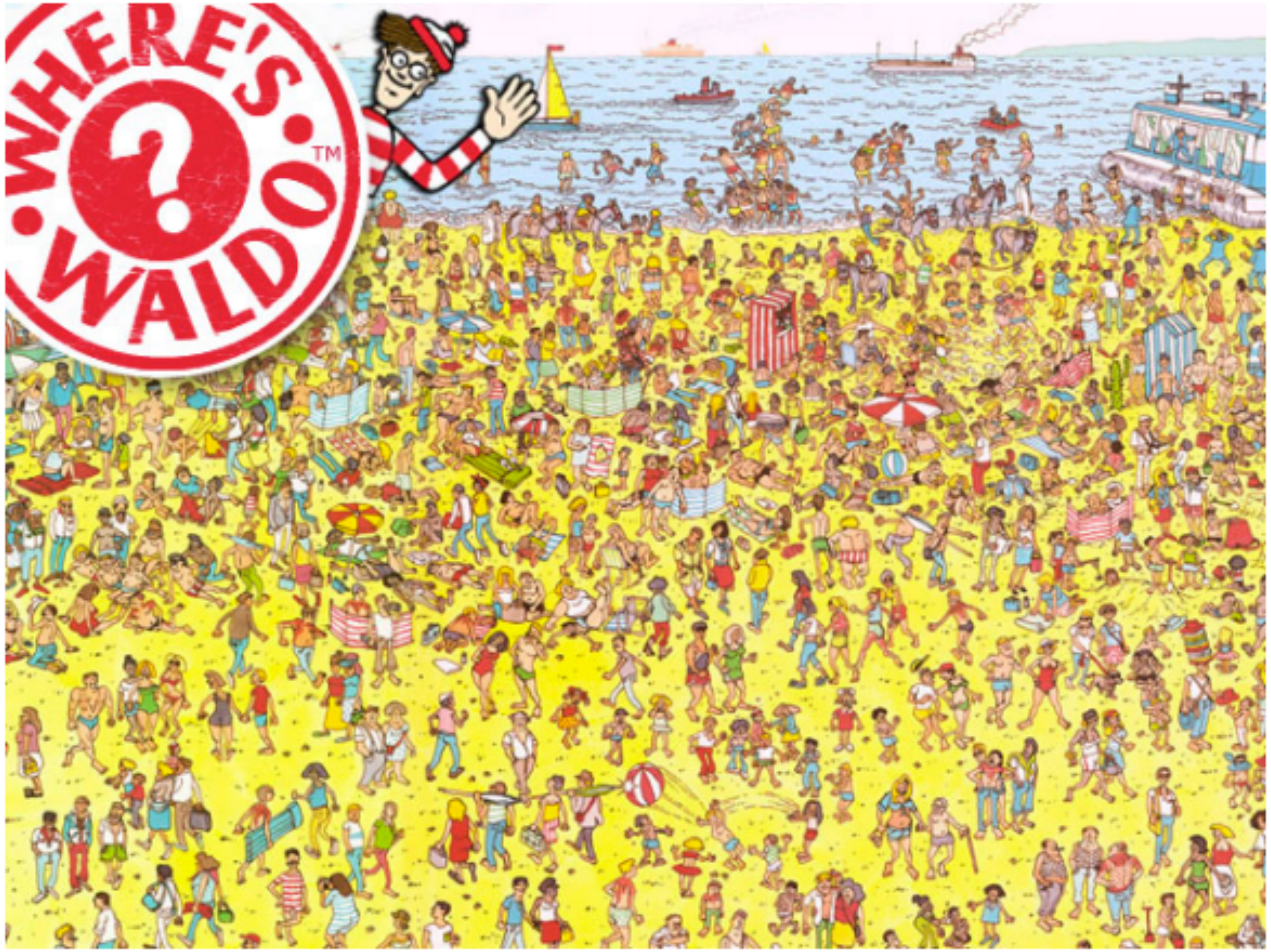


image I

slide credit: Chris McIntosh

1. Make the template as a filter

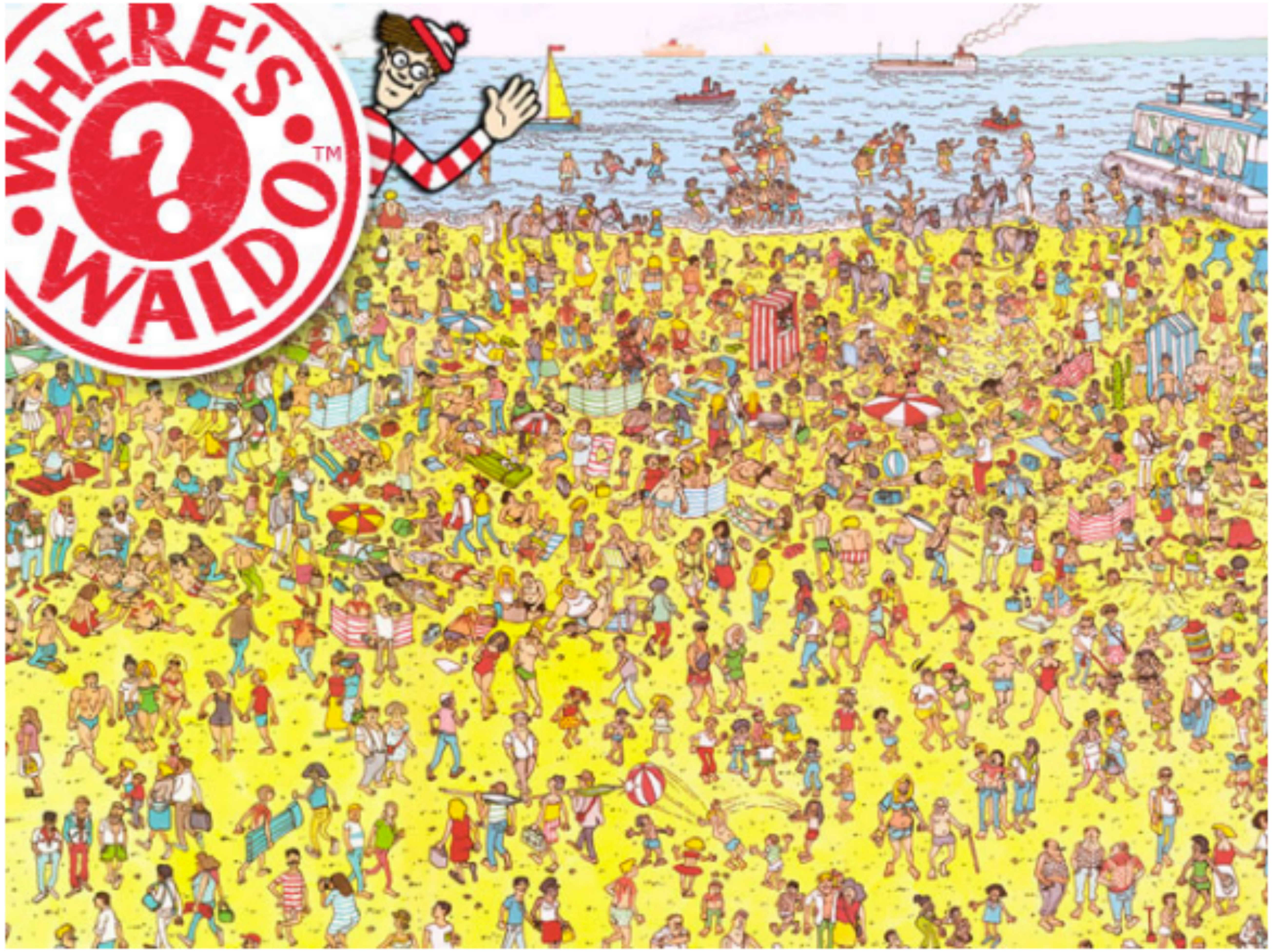
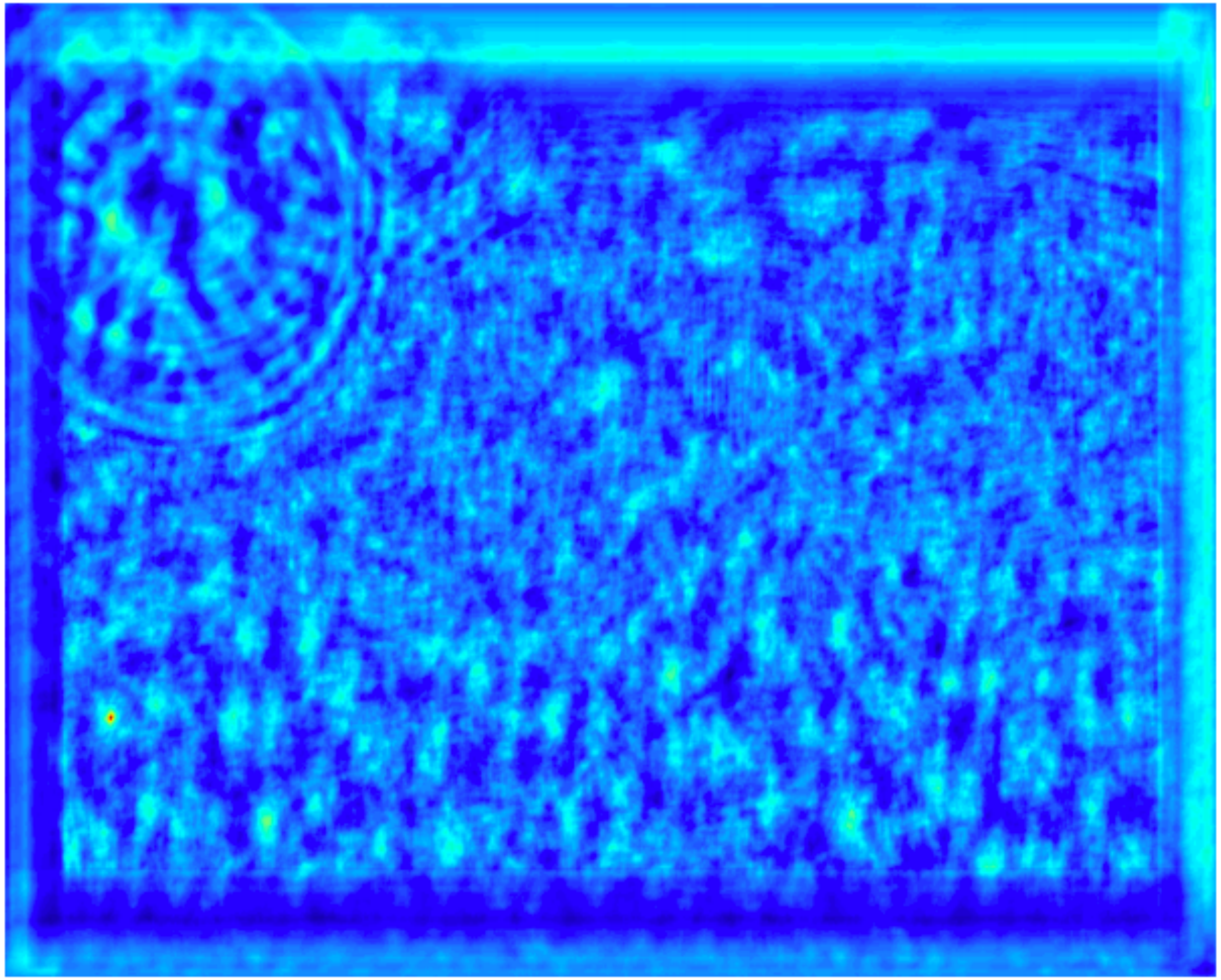


image I

filter F

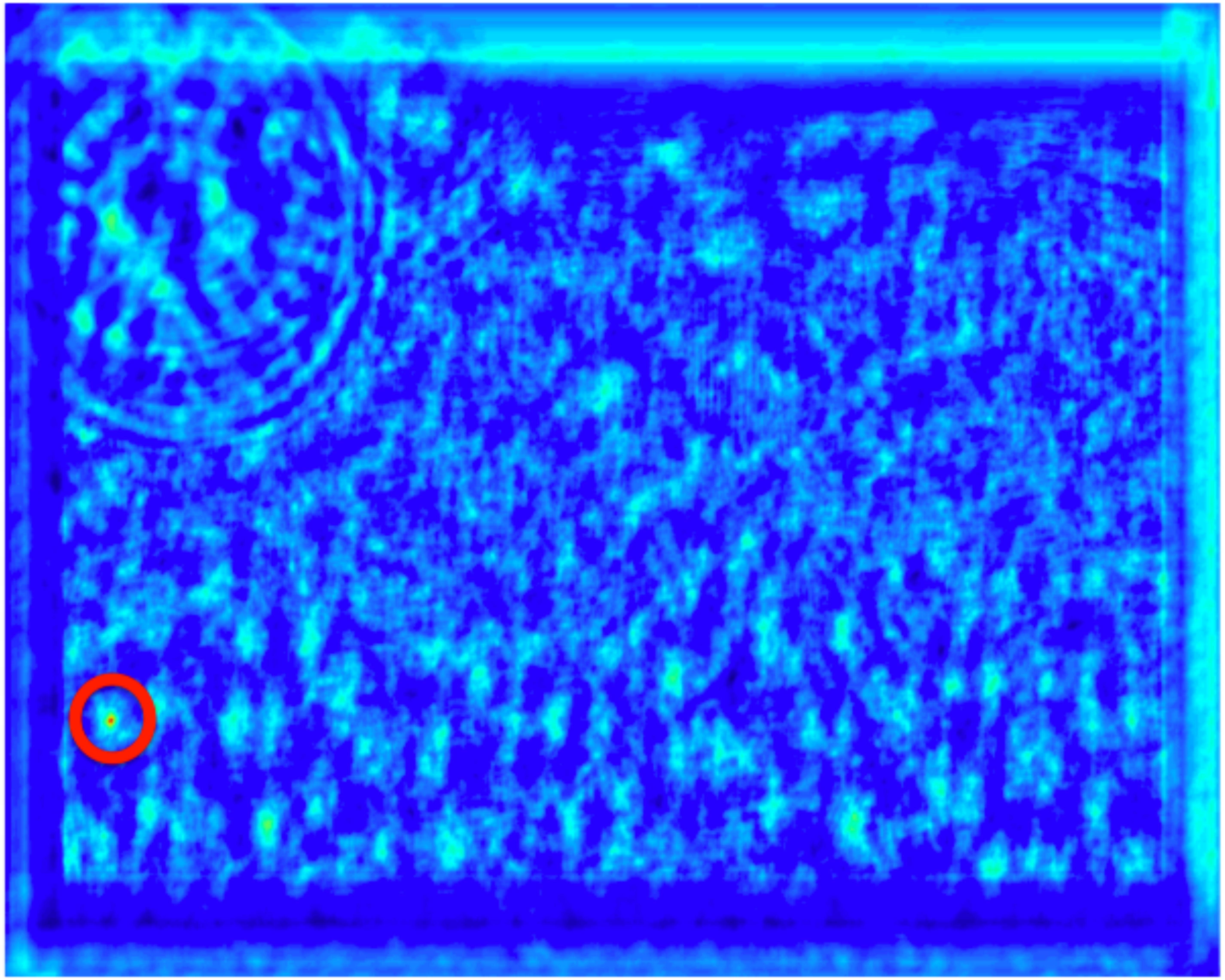
slide credit: Chris McIntosh

2. Result of normalized cross-correlation



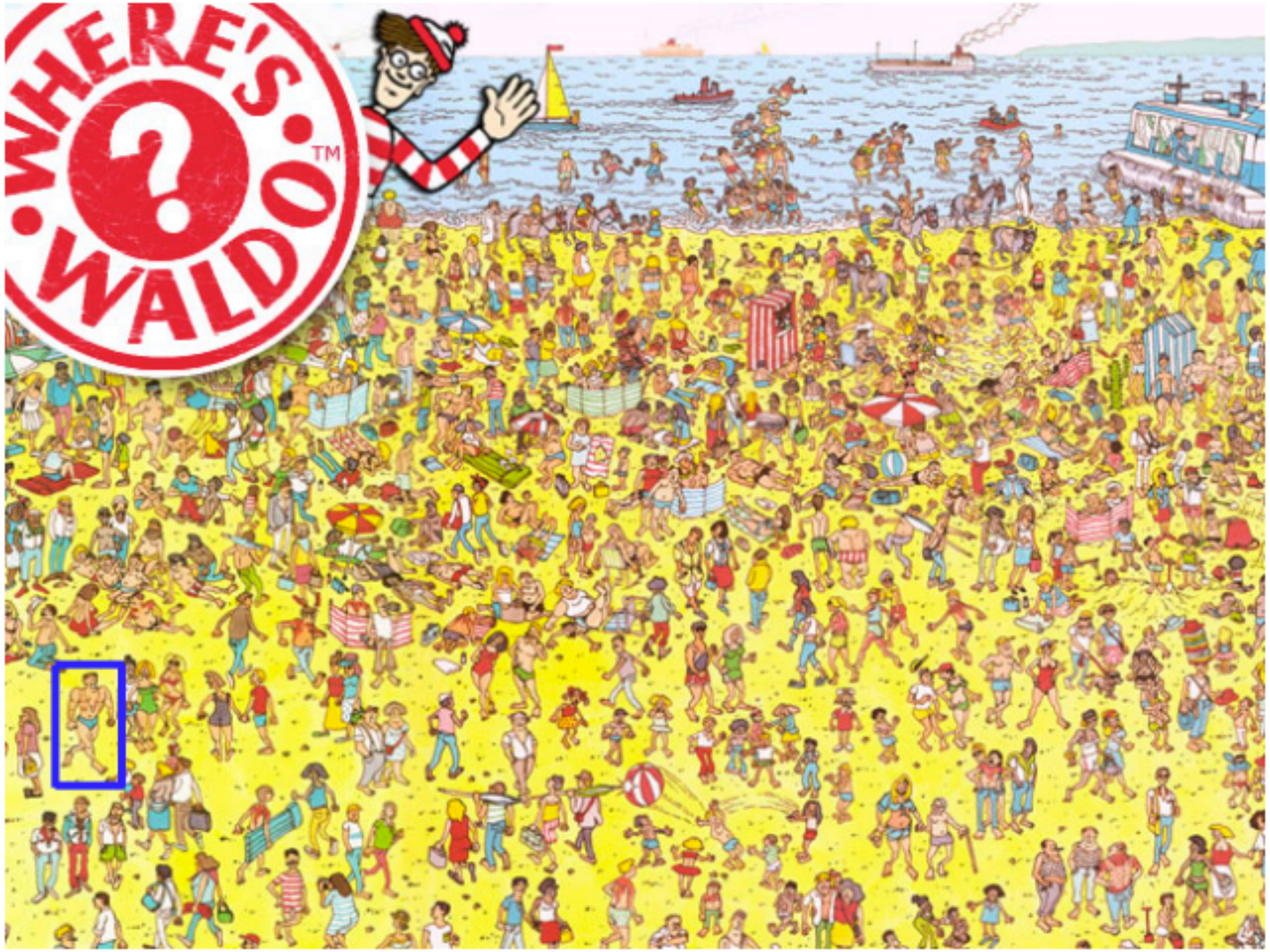
slide credit: Chris McIntosh

3. Find the highest peak



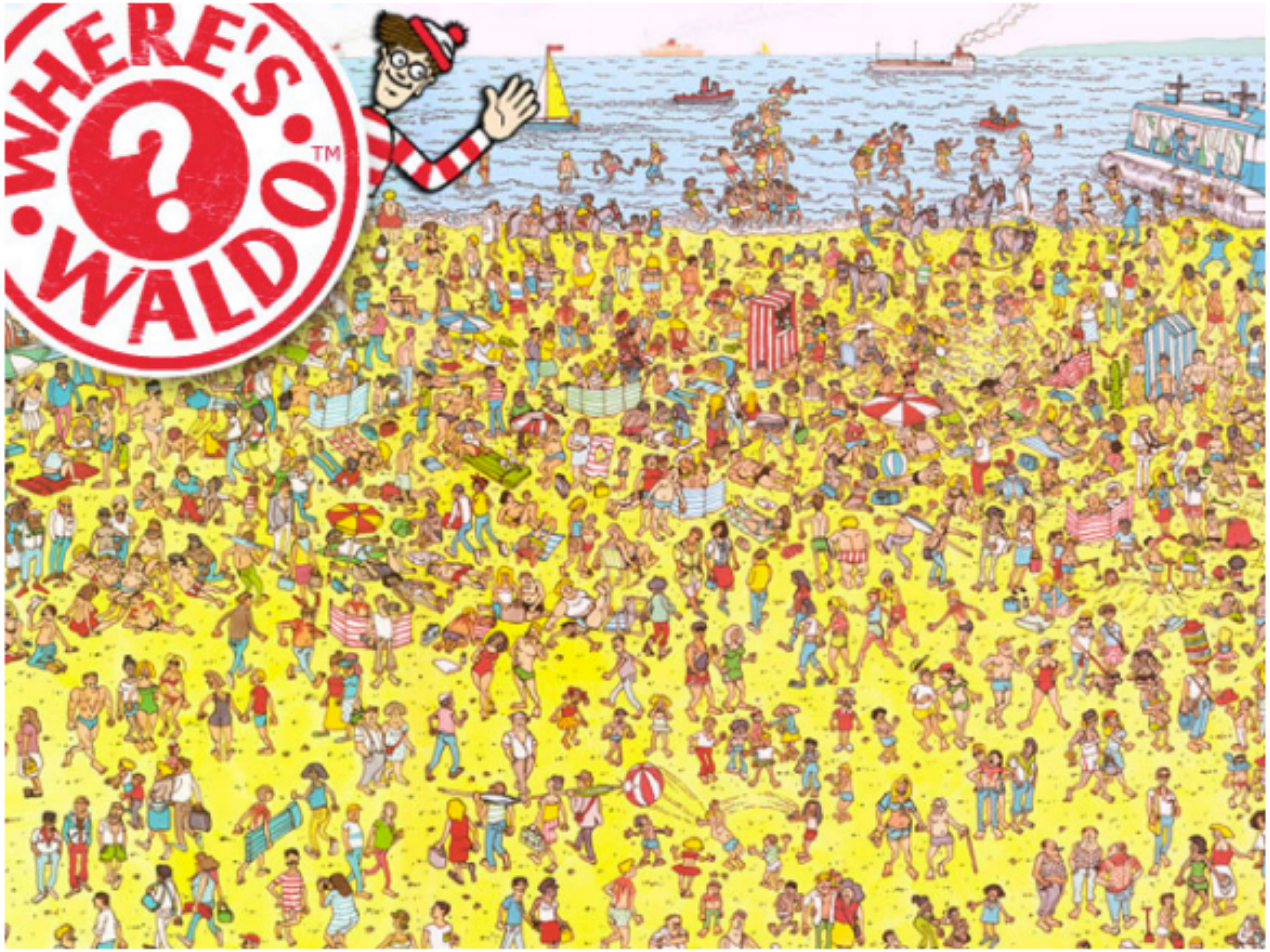
slide credit: Chris McIntosh

4. Put a bounding box (the size of template) at the point



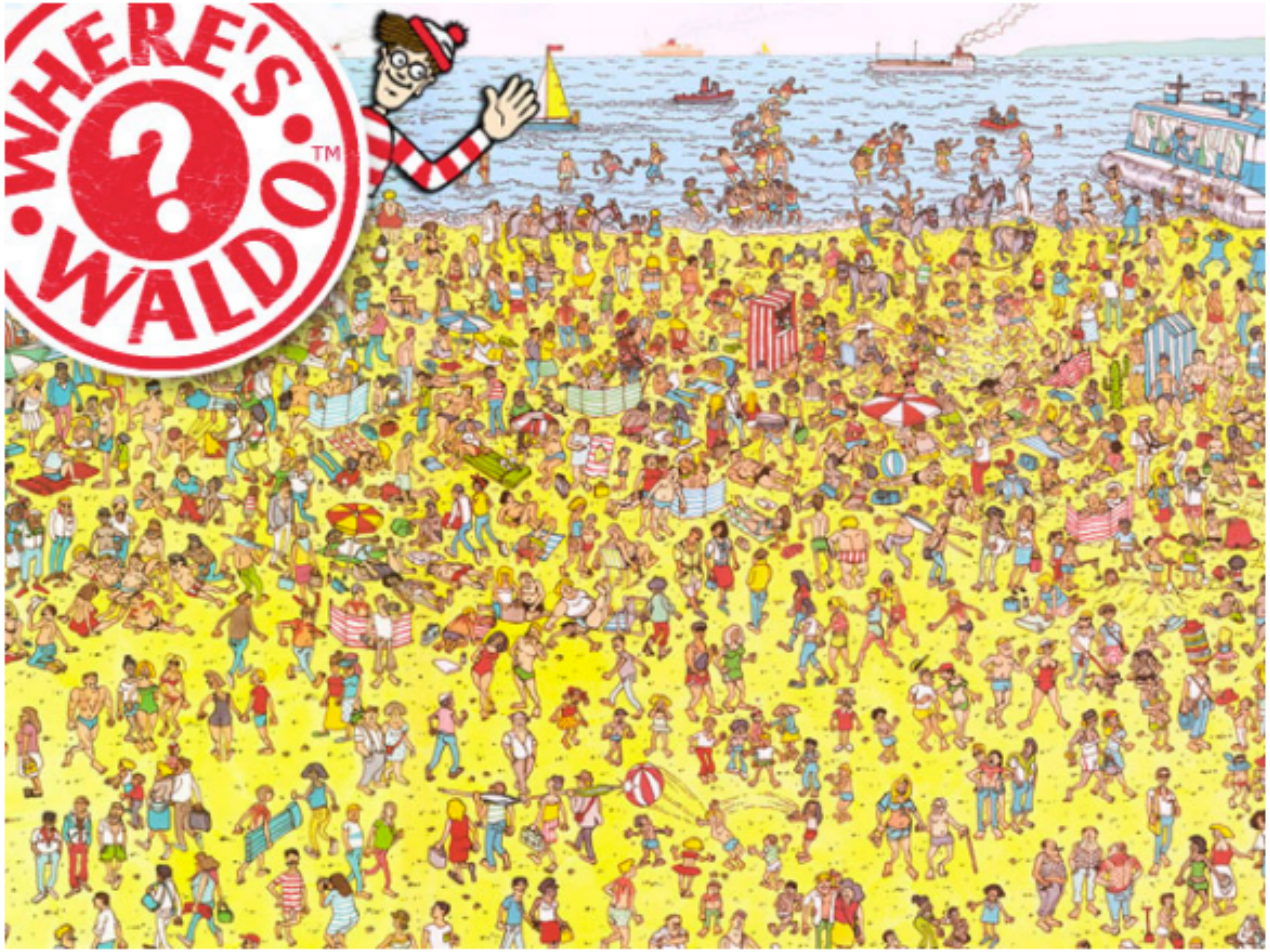
slide credit: Chris McIntosh

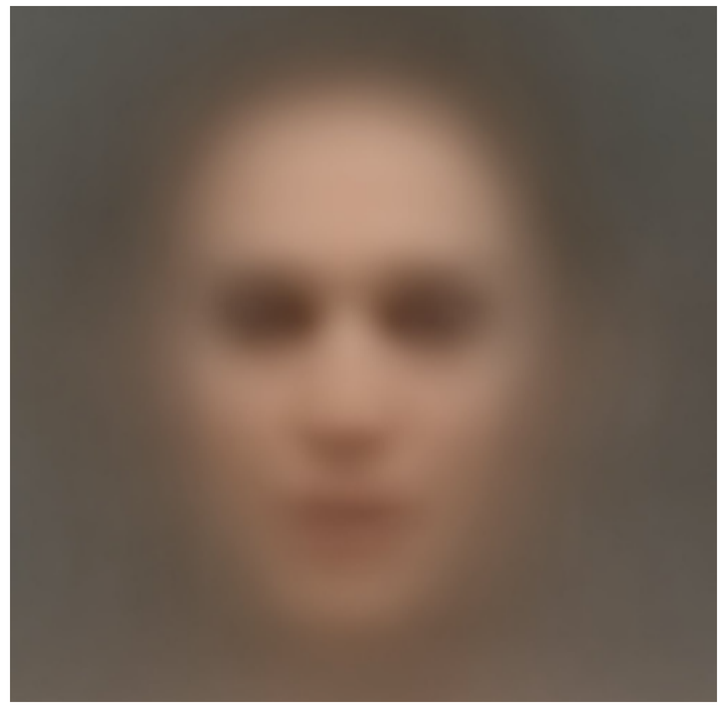
Example 2: Find all persons?



slide credit: Chris McIntosh

Example 2: Find all persons?

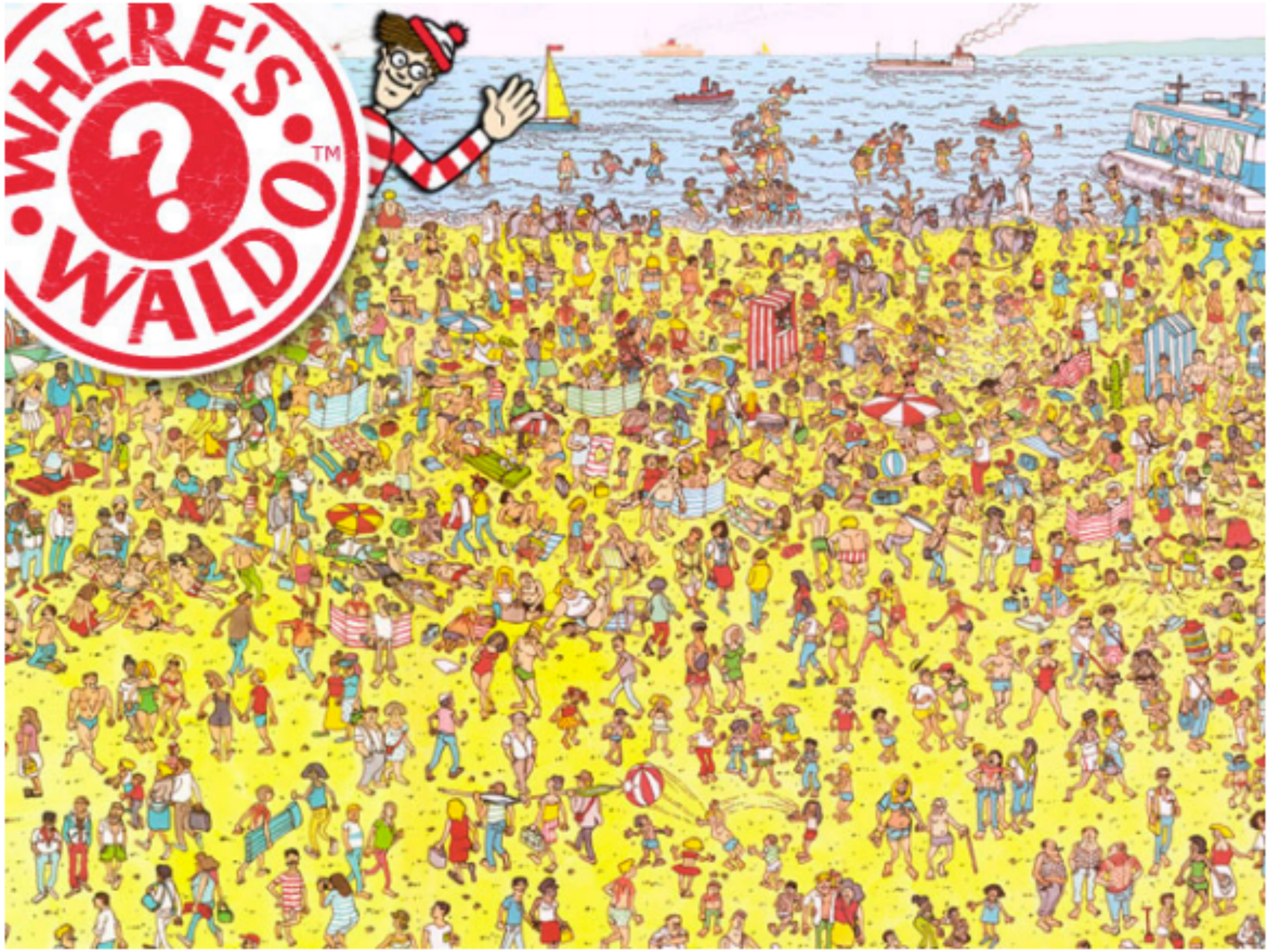


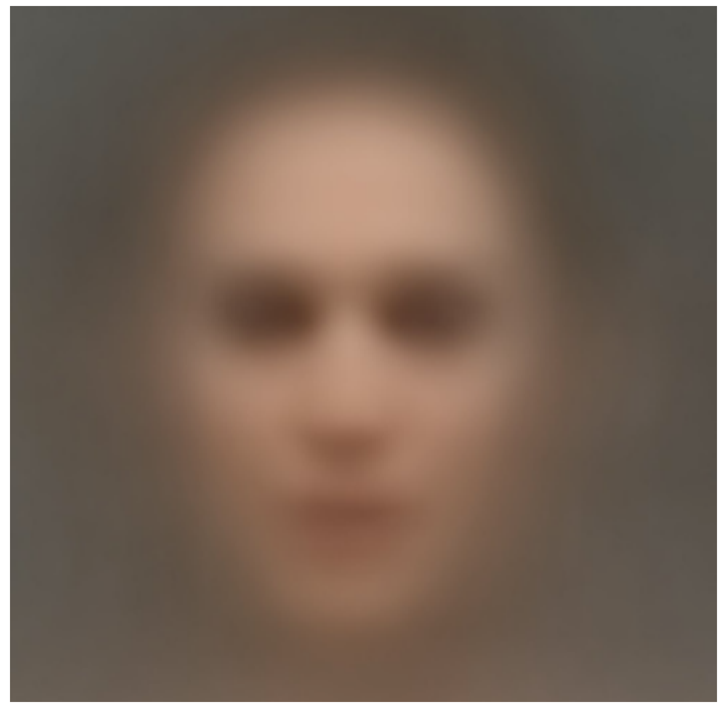


A template for all instances?

slide credit: Chris McIntosh

Example 2: Find all persons?





A template for all instances?

We need features!

slide credit: Chris McIntosh

Object detection

- Introduction
- Pre-CNN time
  - HOG detector
  - Deformable Part-based Model
- CNN time
  - Region CNN
  - Fast versions of RCNN
  - YOLO/SSD
- 3D object detection
- Devil’s in the details

The HOG Detector

N. Dalal and B. Triggs

Histograms of oriented gradients for human detection

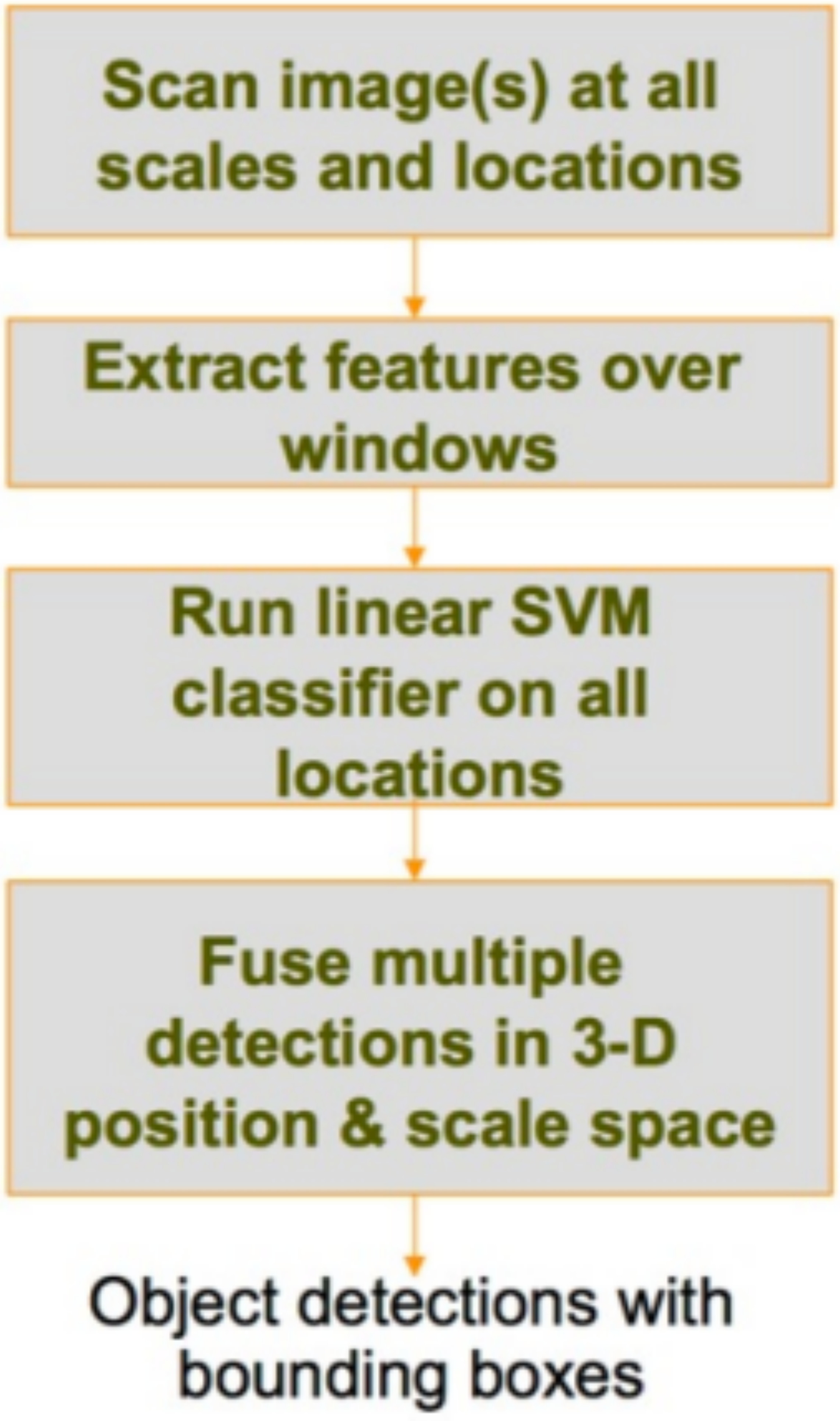
CVPR, 2005

Paper: `http://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf`

cited by 17,502

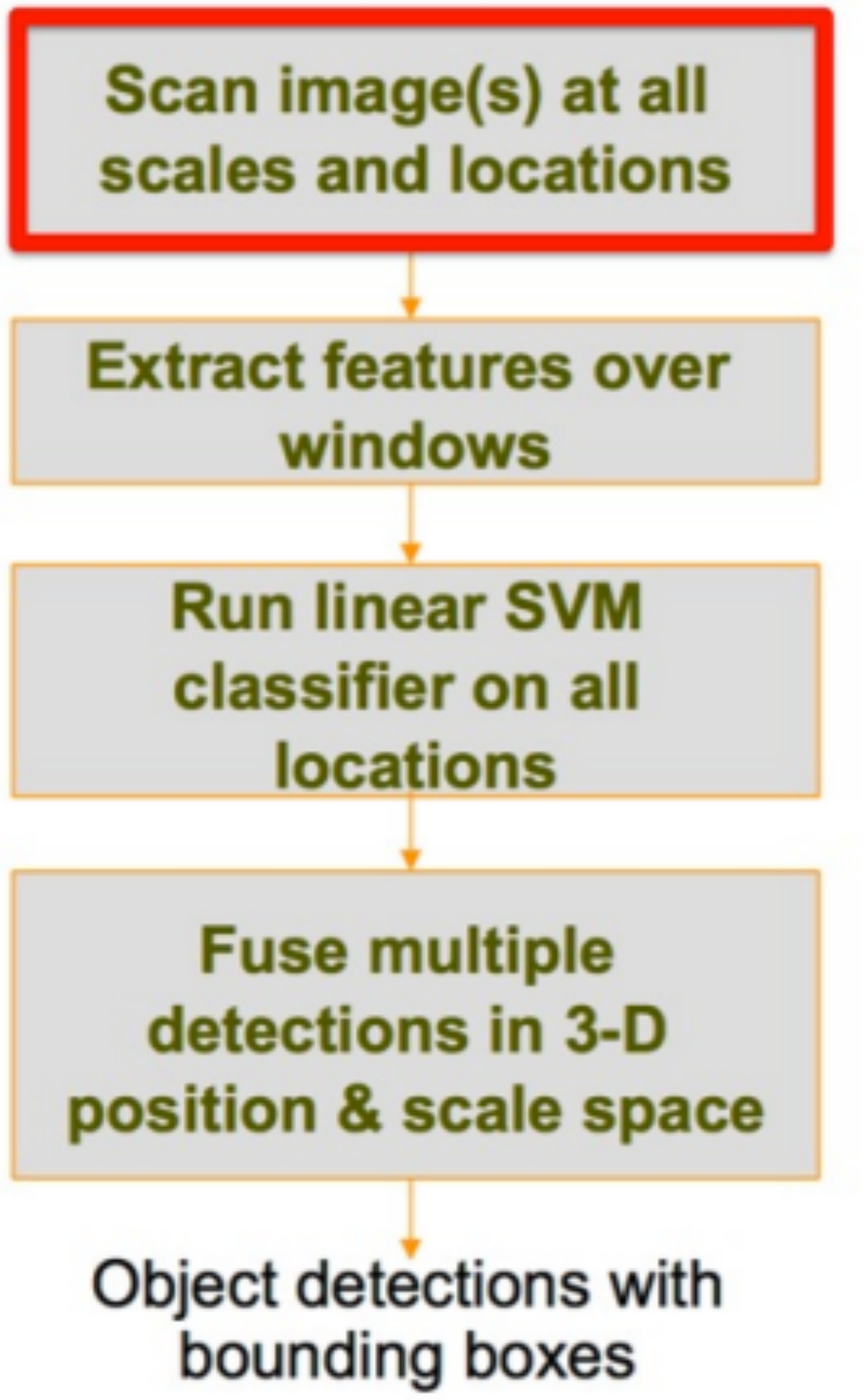
Slide credit: Sanja Fidler

HOG detector: pipeline

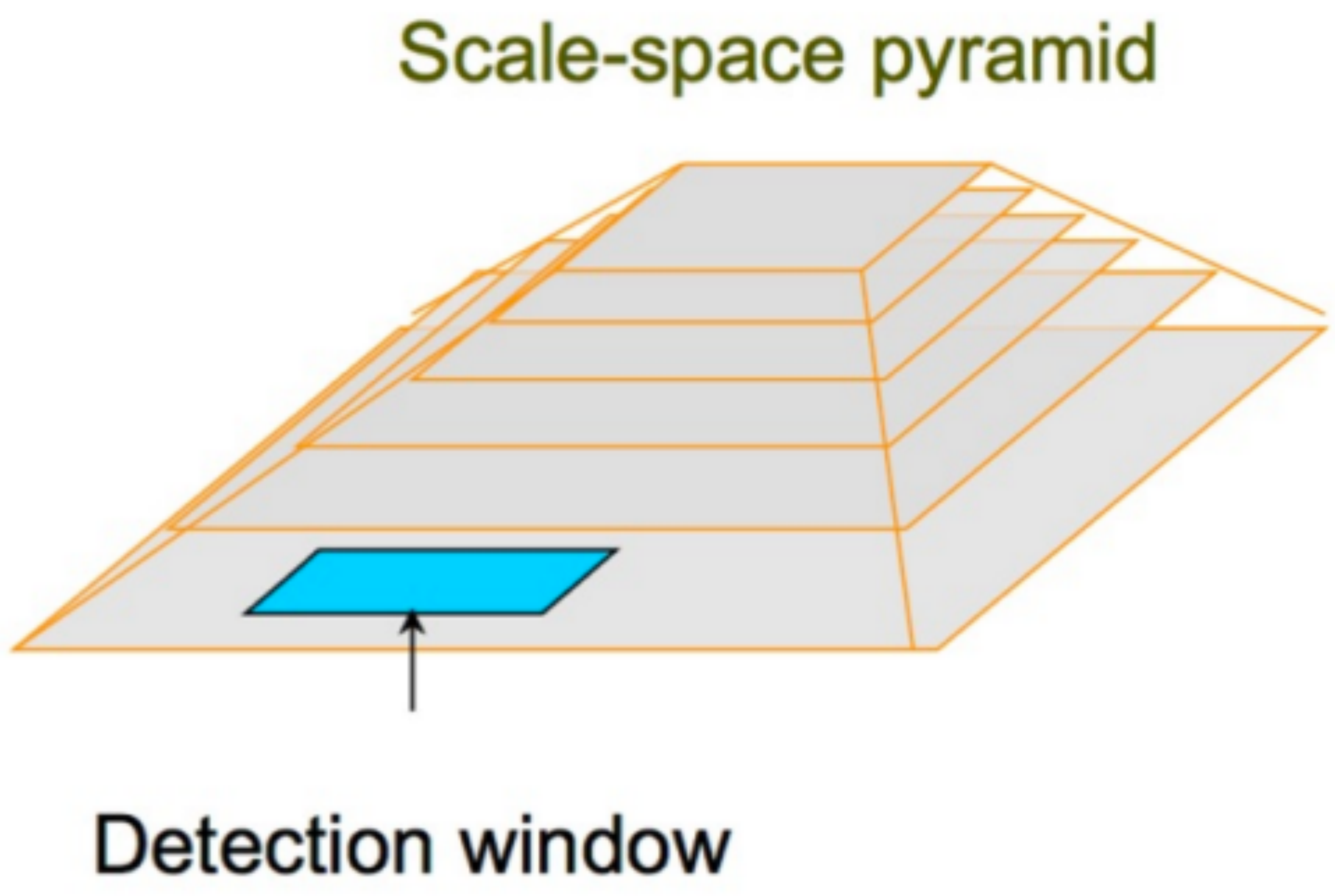


Slide credit: Sanja Fidler

I. Sliding window



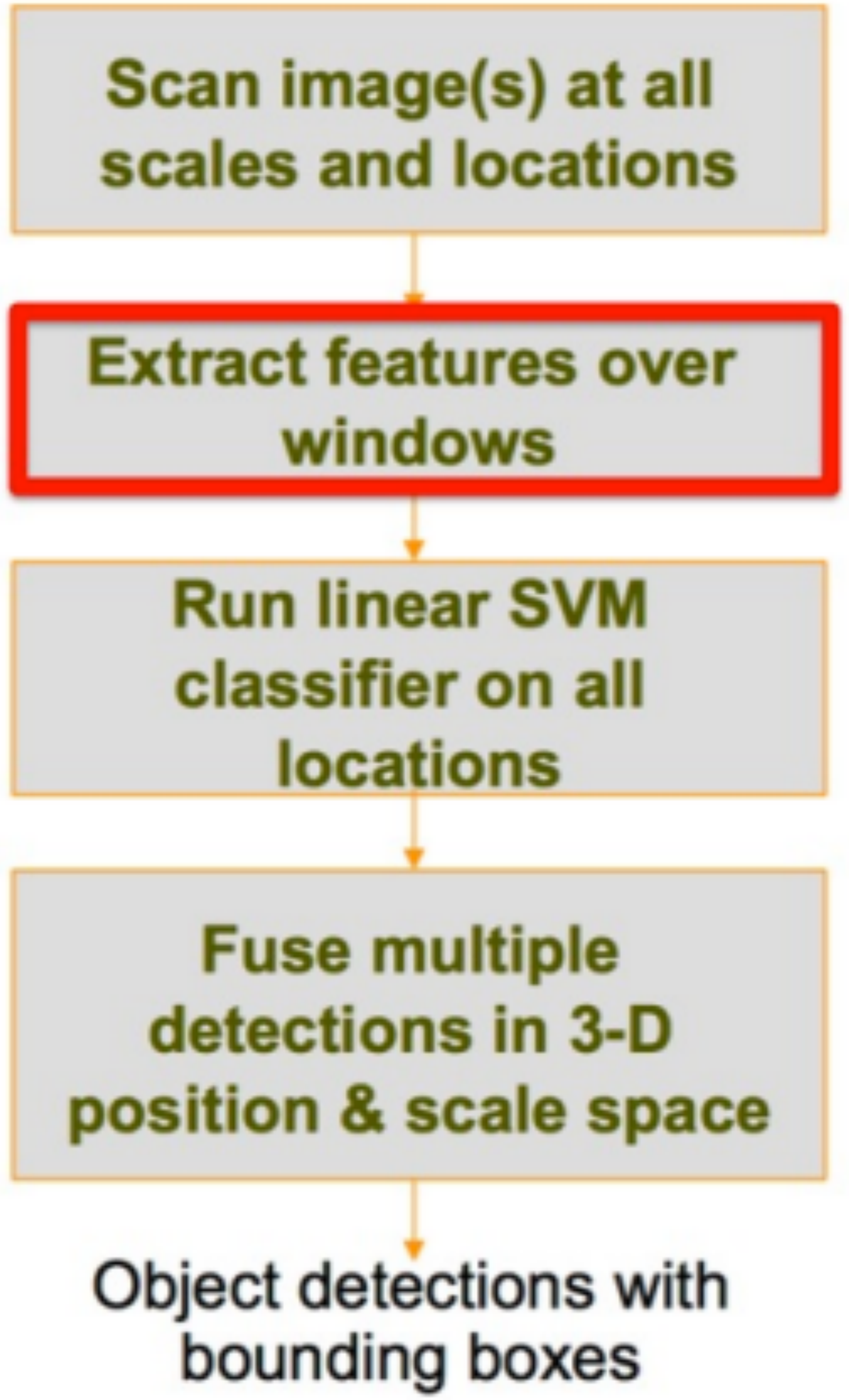
locations

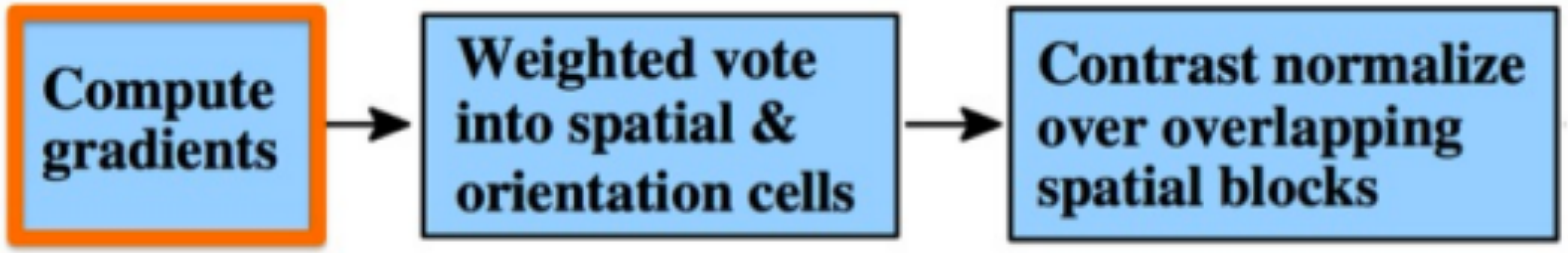


scales

Slide credit: Sanja Fidler

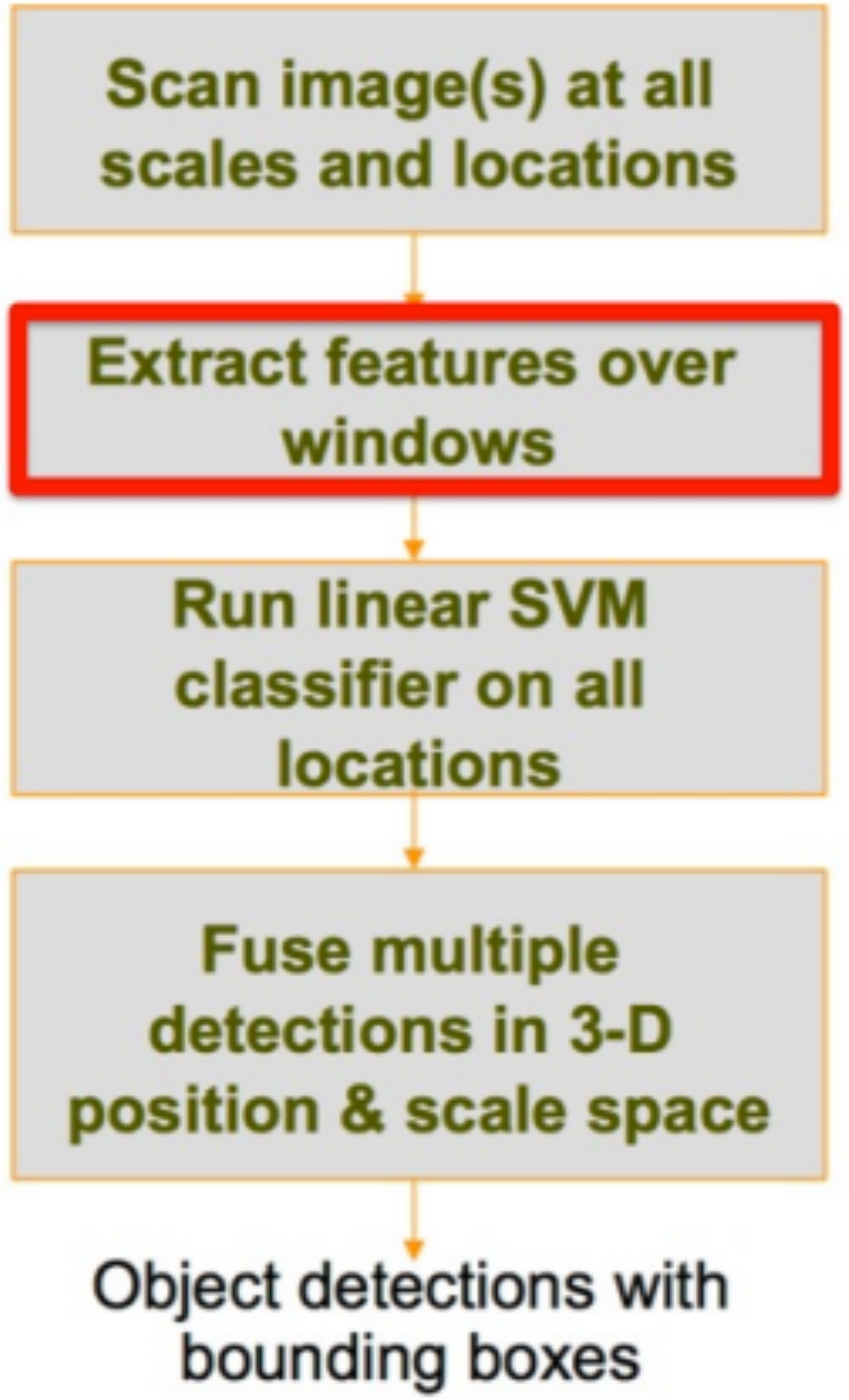
II. Histograms of Oriented Gradients

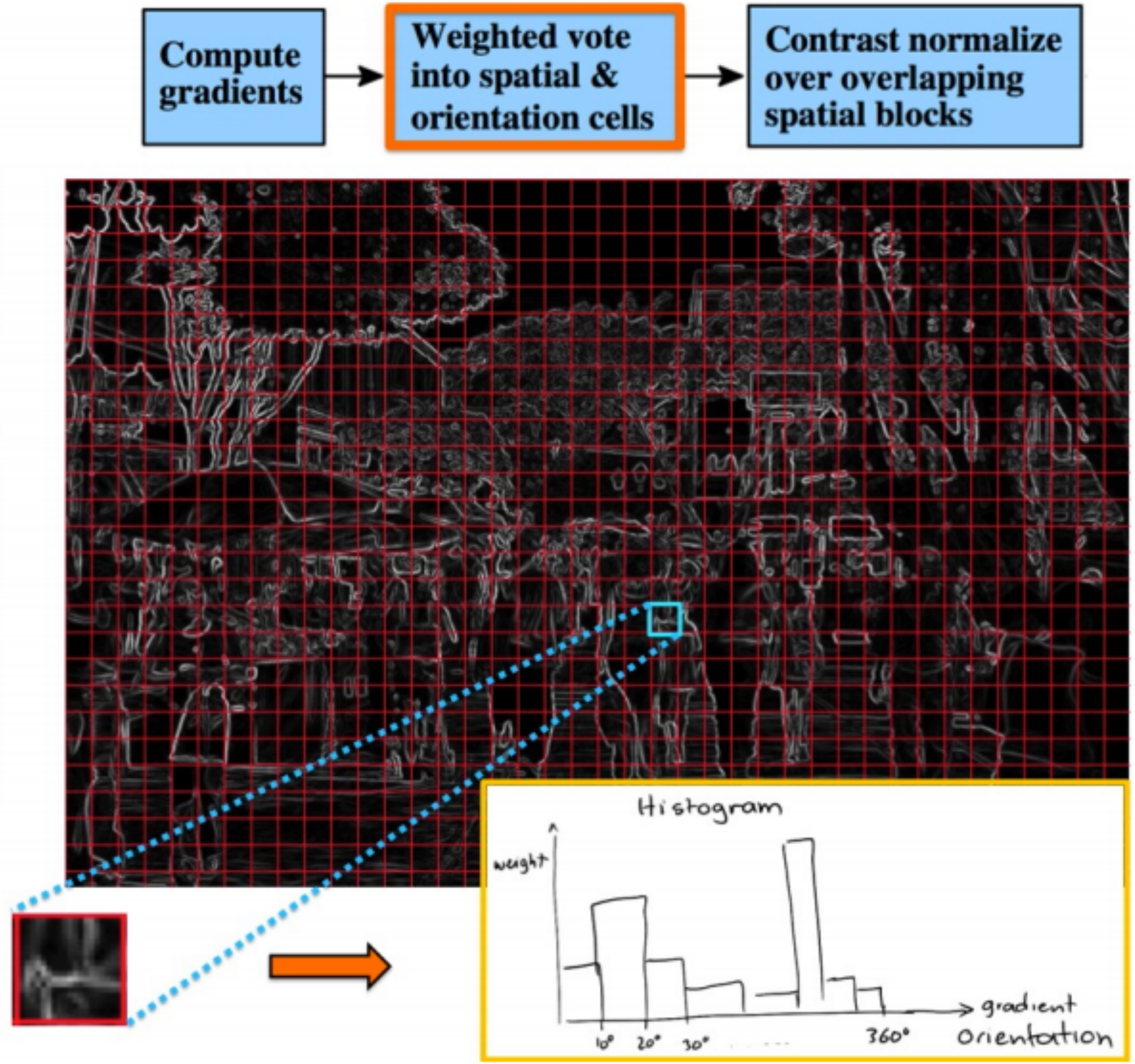




Slide credit: Sanja Fidler

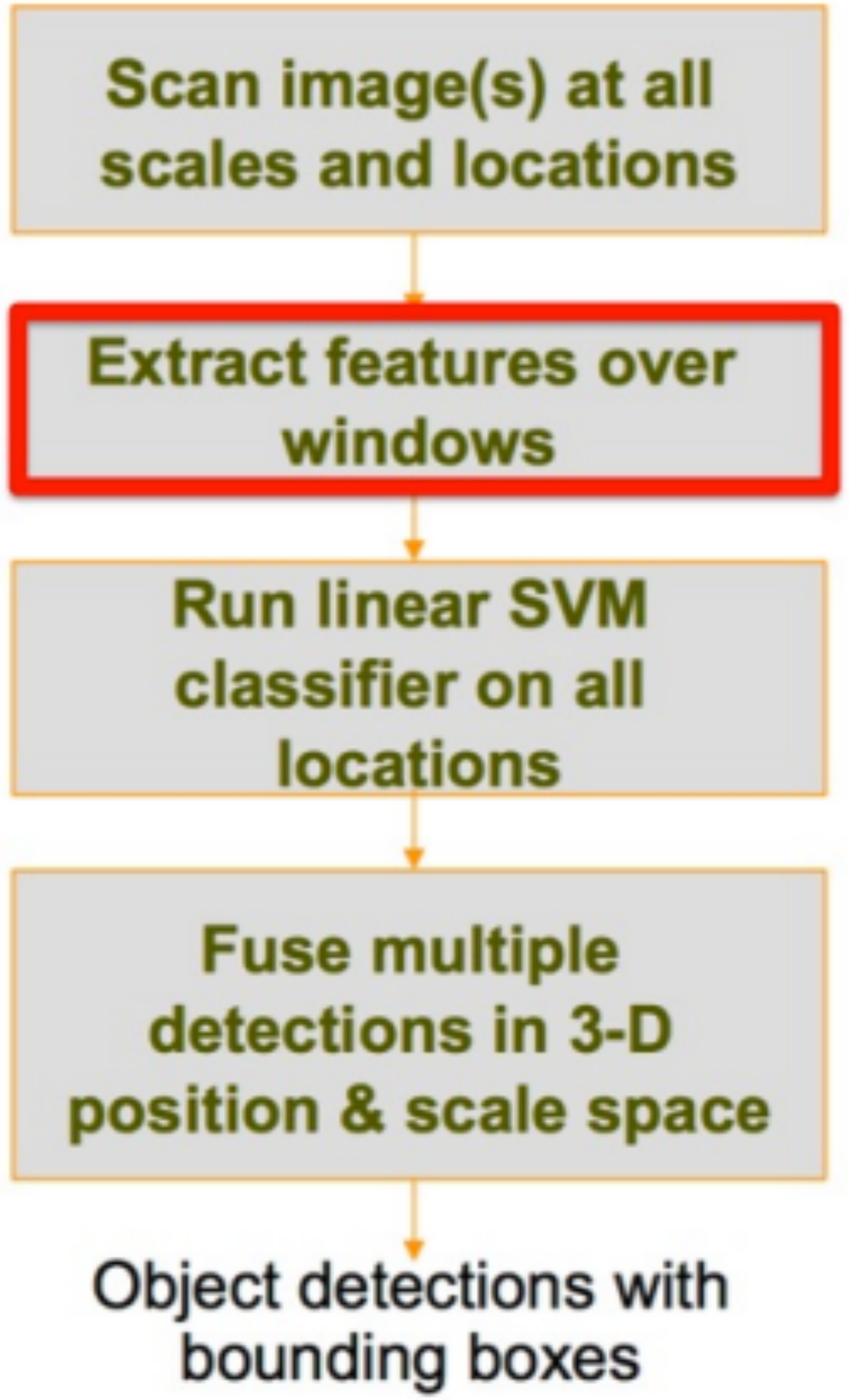
II. Histograms of Oriented Gradients

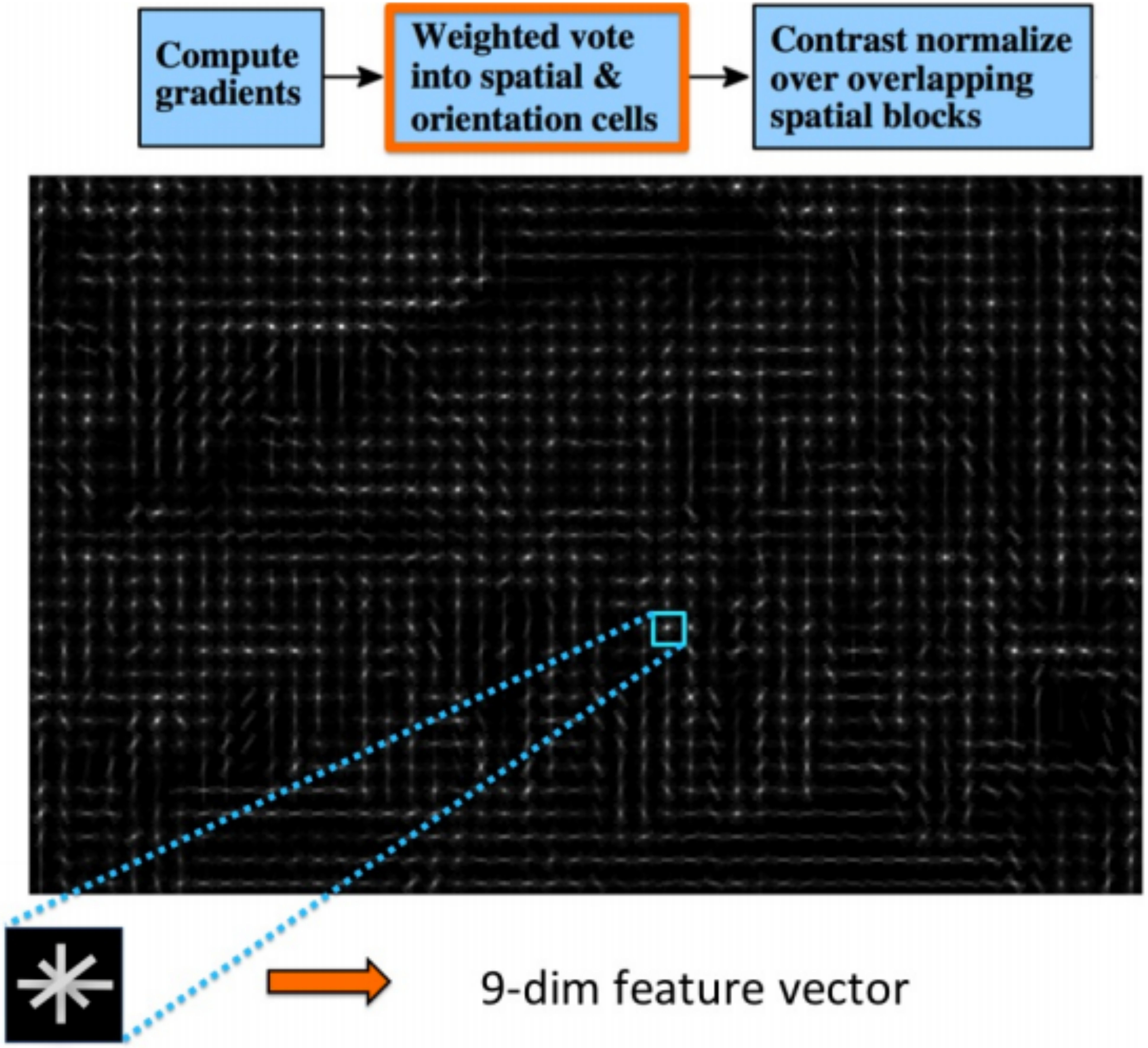




Slide credit: Sanja Fidler

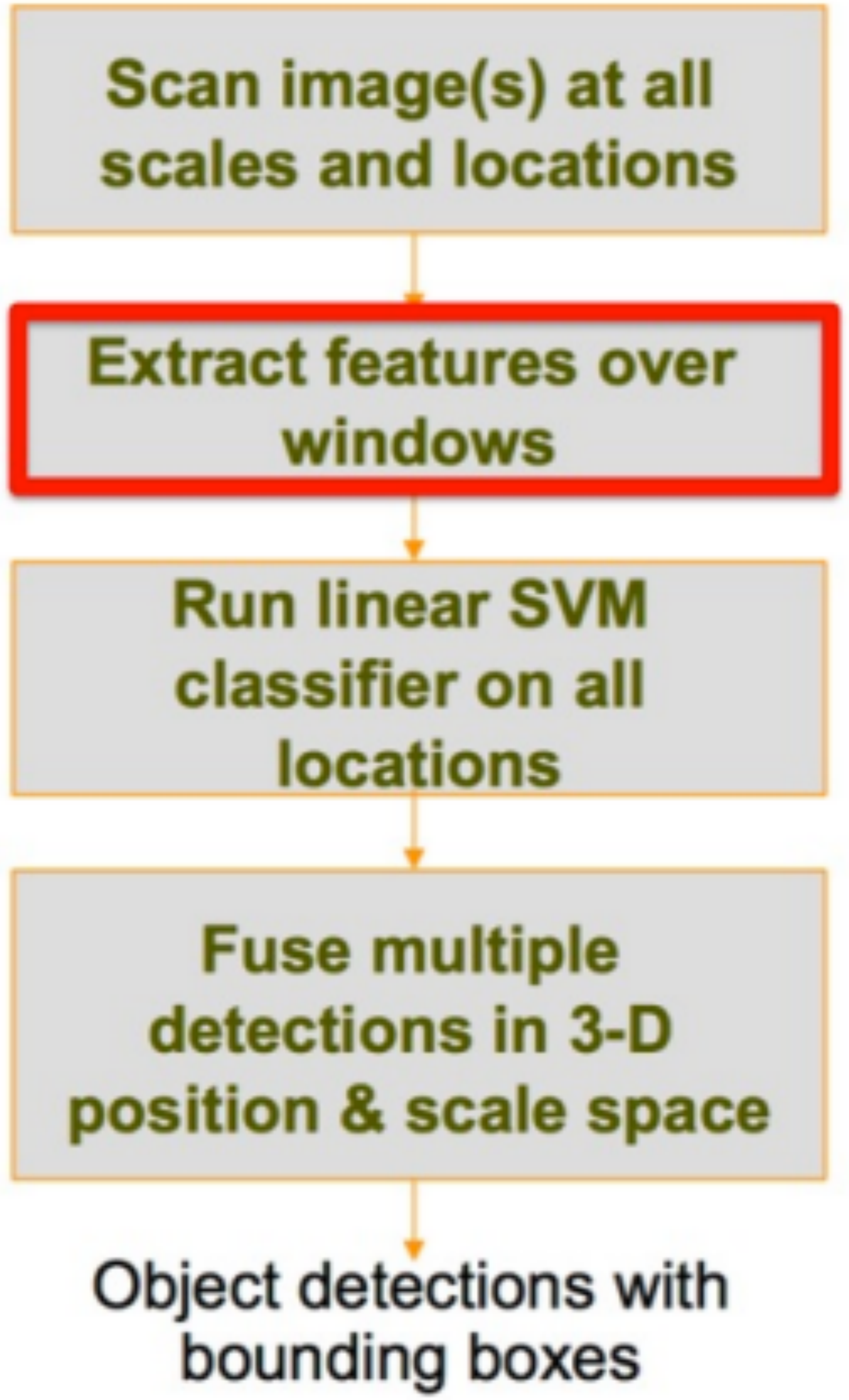
II. Histograms of Oriented Gradients

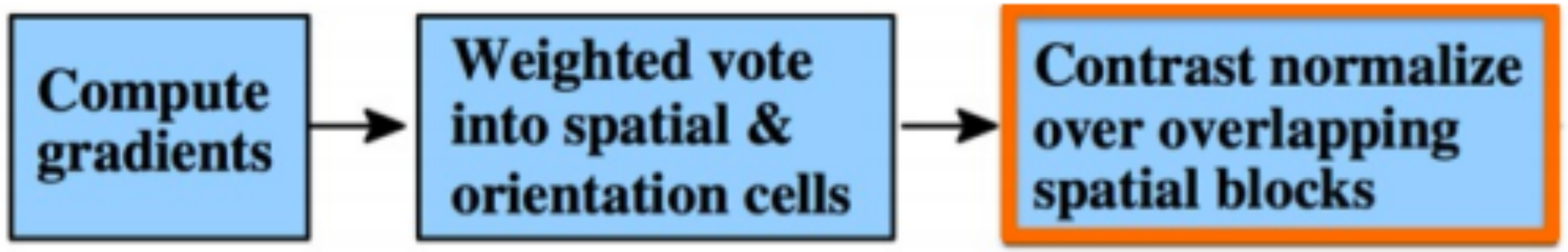


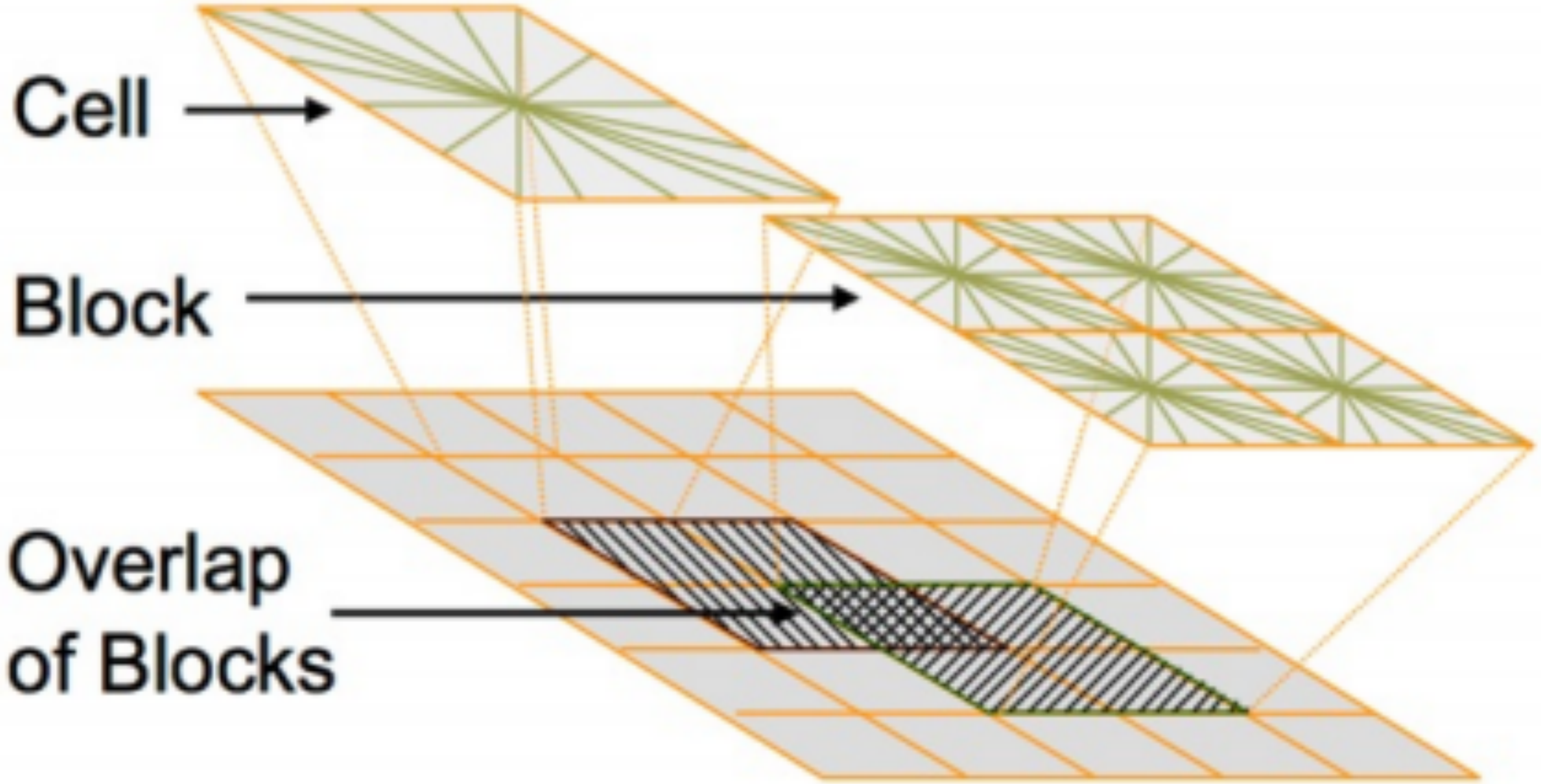


Slide credit: Sanja Fidler

II. Histograms of Oriented Gradients



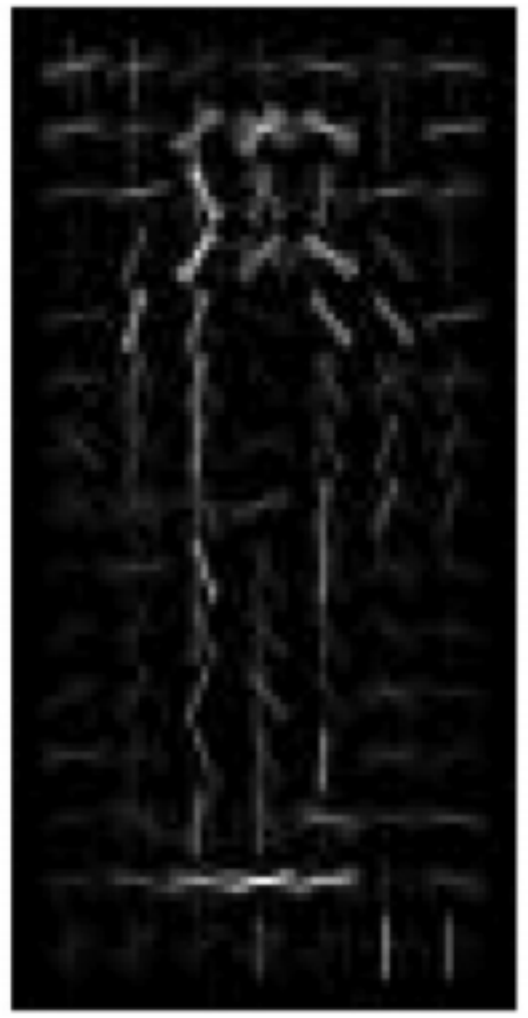




Feature vector f = [..., ..., ...]

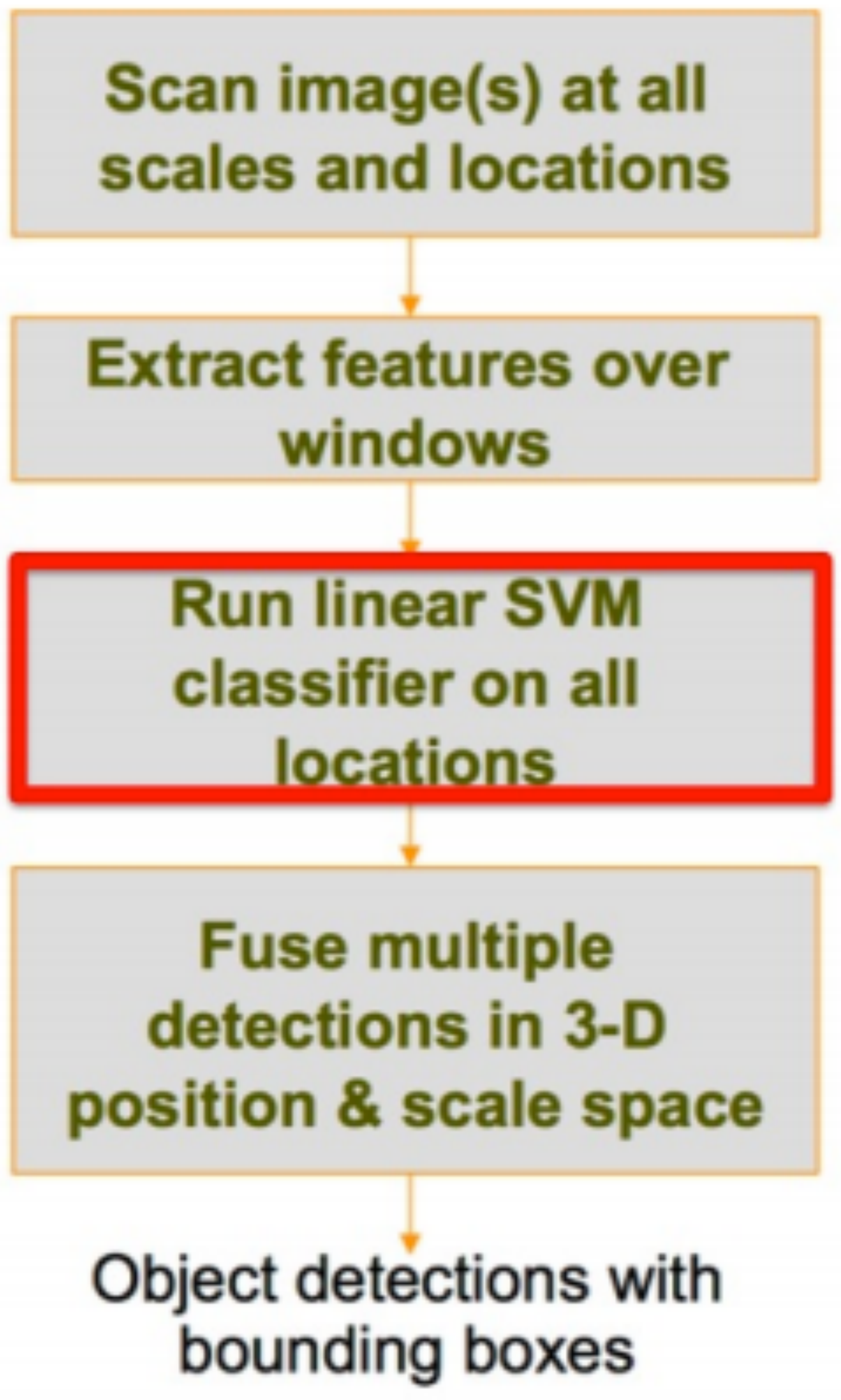
L2 normalization in each block:

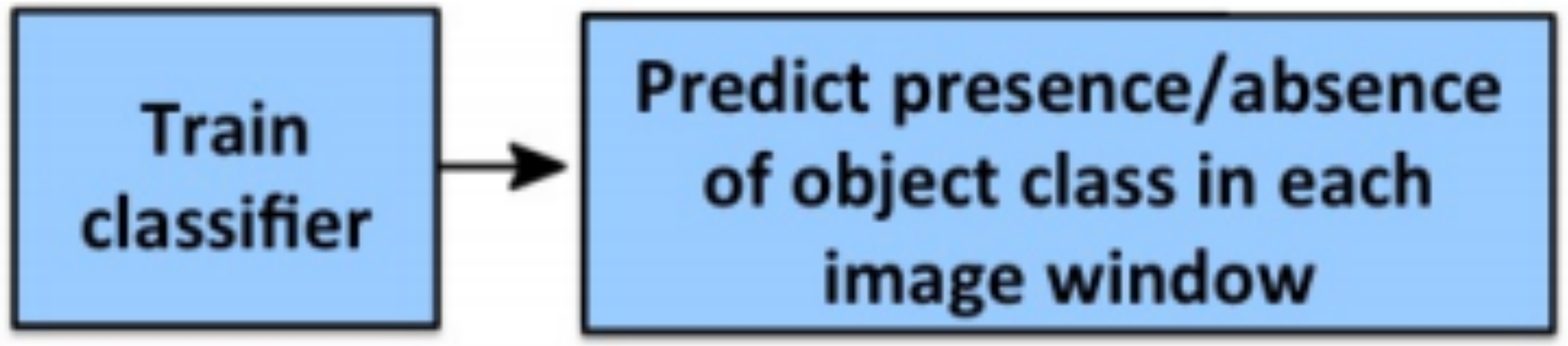
$$\mathbf{f} = \frac{\mathbf{f}}{\sqrt{||\mathbf{f}||_2^2 + \epsilon^2}}$$



Slide credit: Sanja Fidler

III. SVM classifier





Training:

- **Train** a classifier (eg, person vs no person)

Detection:

- Use the trained classifier to **predict** presence/absence of object class in each window in the image

Slide credit: Sanja Fidler

III. SVM classifier - training

Train classifier → **Predict presence/absence of object class in each image window**

Learning phase

Input: Annotations on training images

↓

Create fixed-resolution normalised training image data set

↓

Encode images into feature spaces

↓

Learn binary classifier

positive training examples

negative training examples

- All image crops **are scaled to the same size** (for this example (15x8) x (7x8) pixels), where 8 is the width/height of each HOG cell in pixels
- **Cool trick**: take a bigger region than each annotated object to also capture **context** (works better!)

Pics: S. Lazebnik

Slide credit: Sanja Fidler

III. SVM classifier - training

Train classifier → Predict presence/absence of object class in each image window

Learning phase

Input: Annotations on training images

↓

Create fixed-resolution normalised training image data set

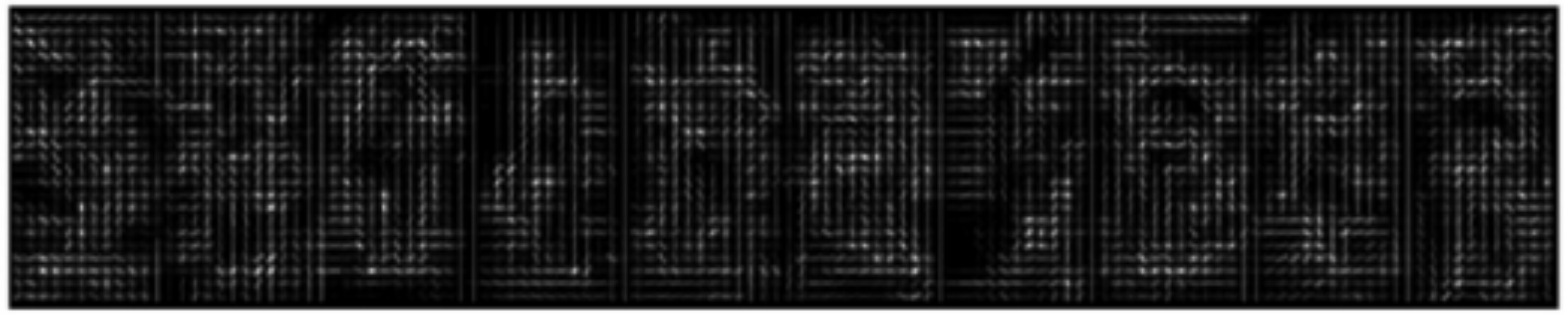
↓

Encode images into feature spaces

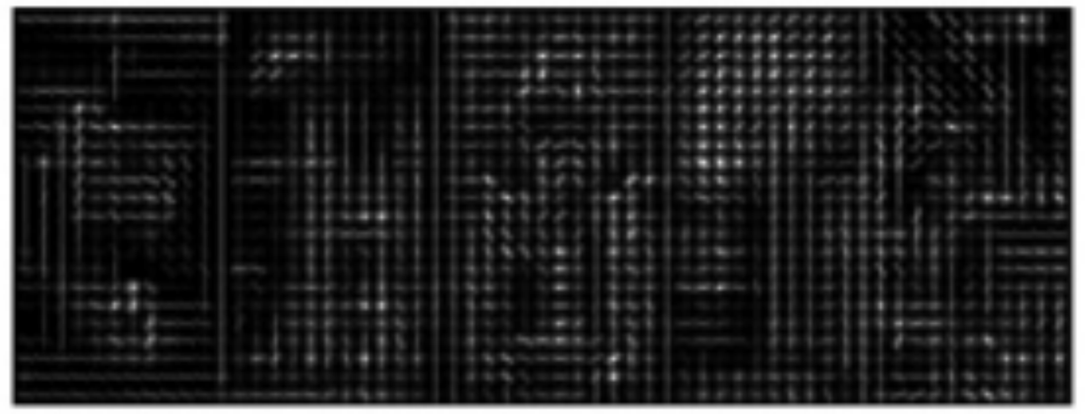
↓

Learn binary classifier

positive training examples



negative training examples

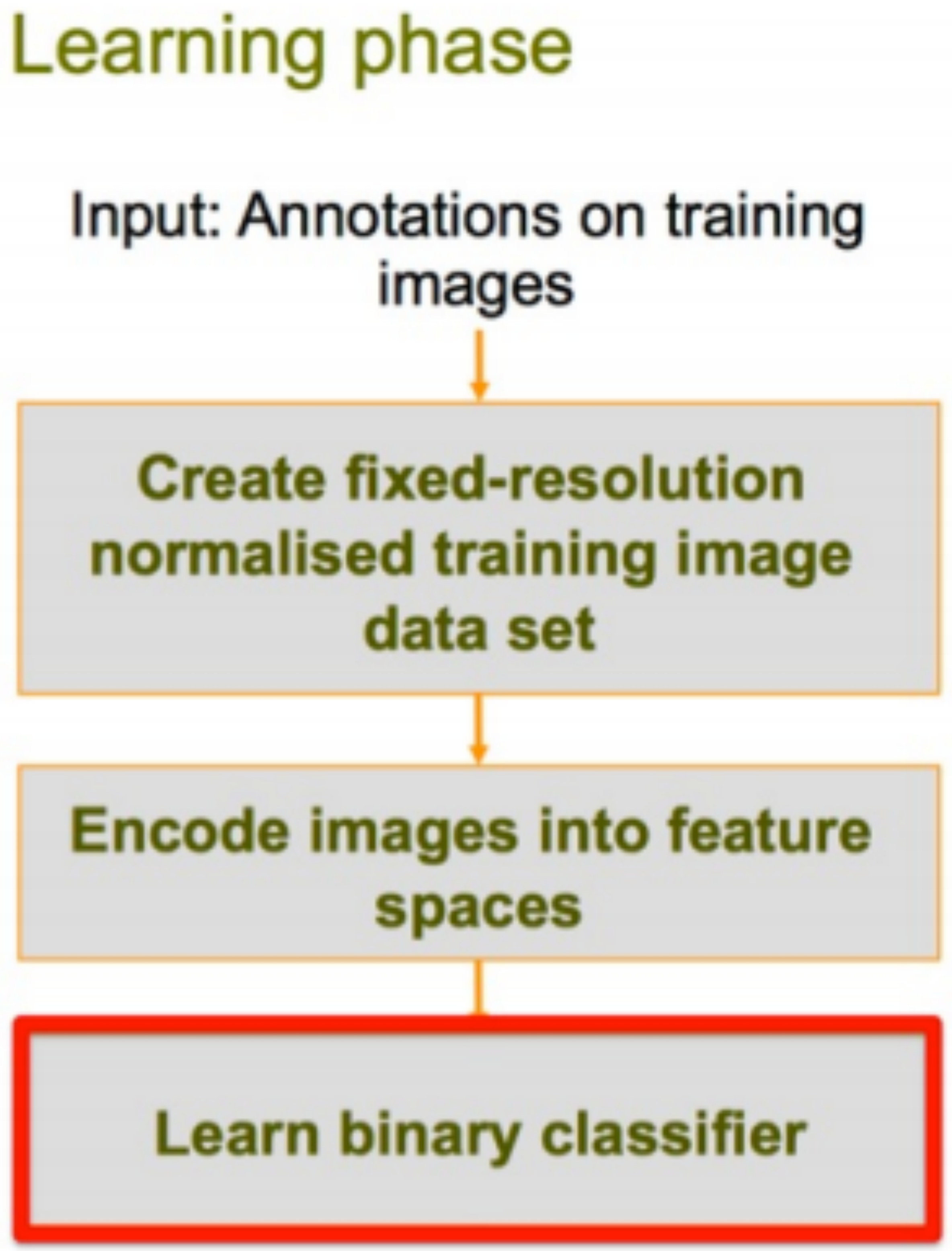


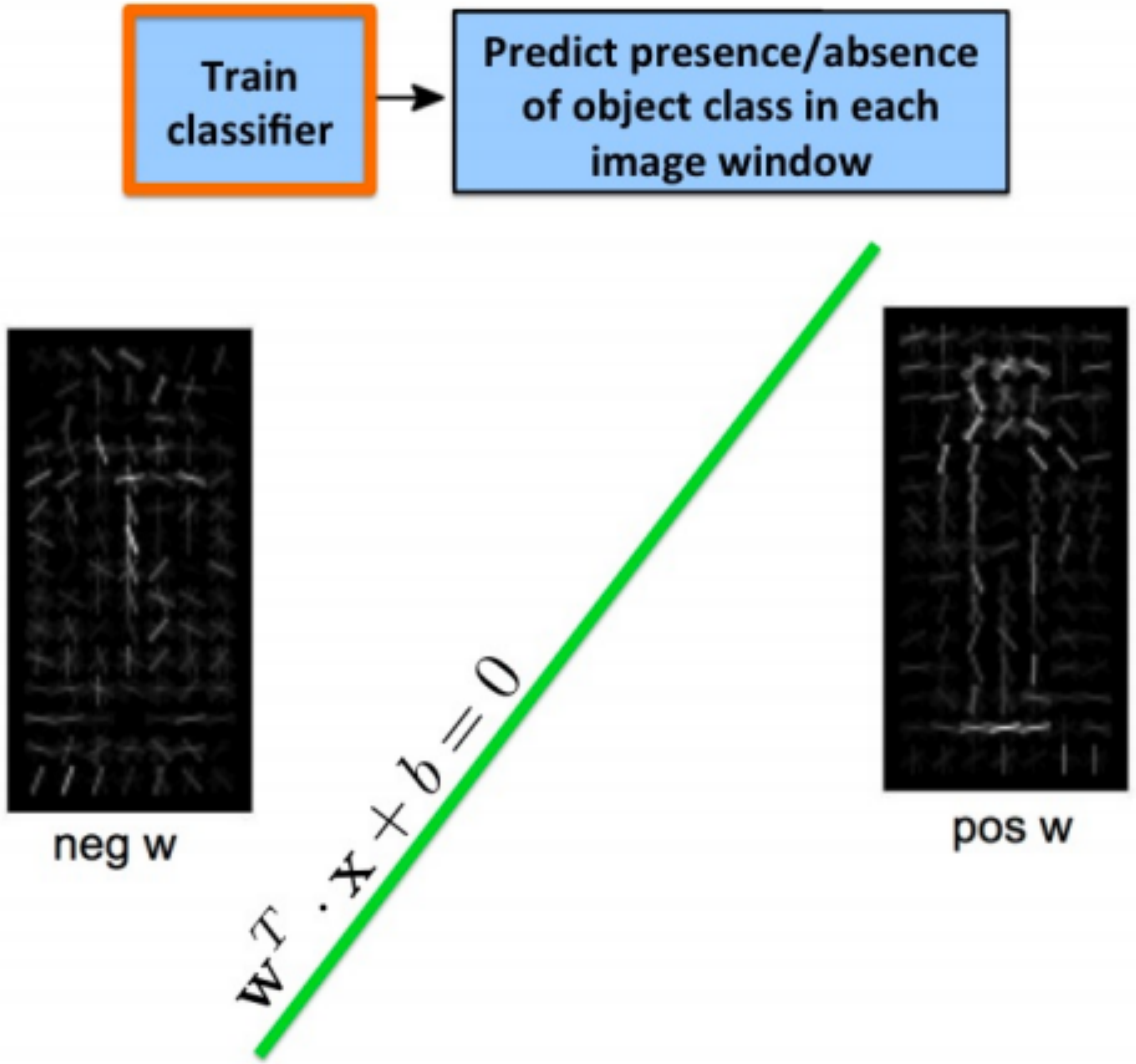
*** These are just feature visualizations. Each \`\`picture'' is really a 15x7x31 feature matrix.

Before training a classifier, we vectorize each of these examples: f=f(:)

Slide credit: Sanja Fidler

III. SVM classifier - training

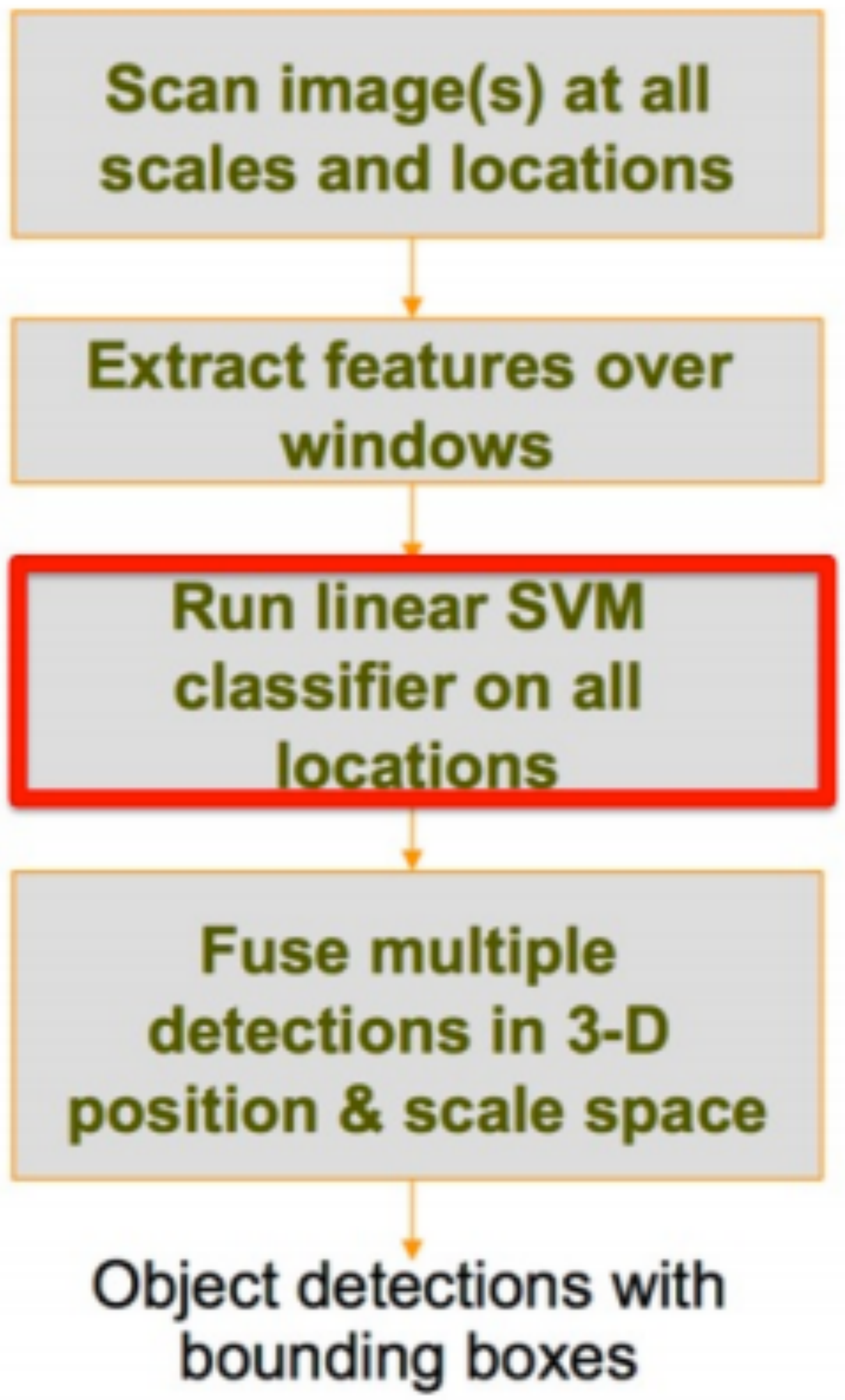




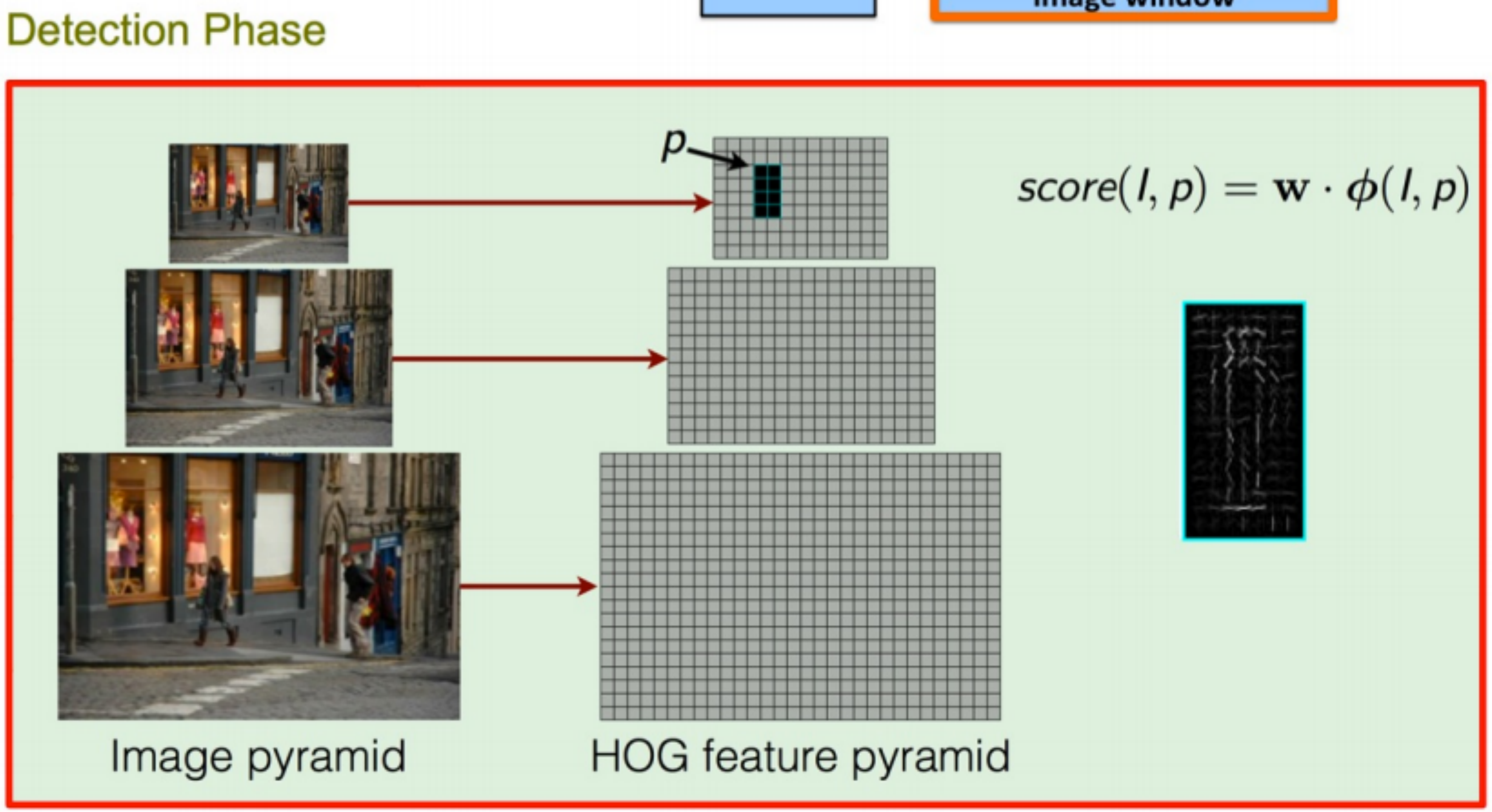
Train classifier. SVM (Support Vector Machines) is typically used.

Slide credit: Sanja Fidler

III. SVM classifier - detection



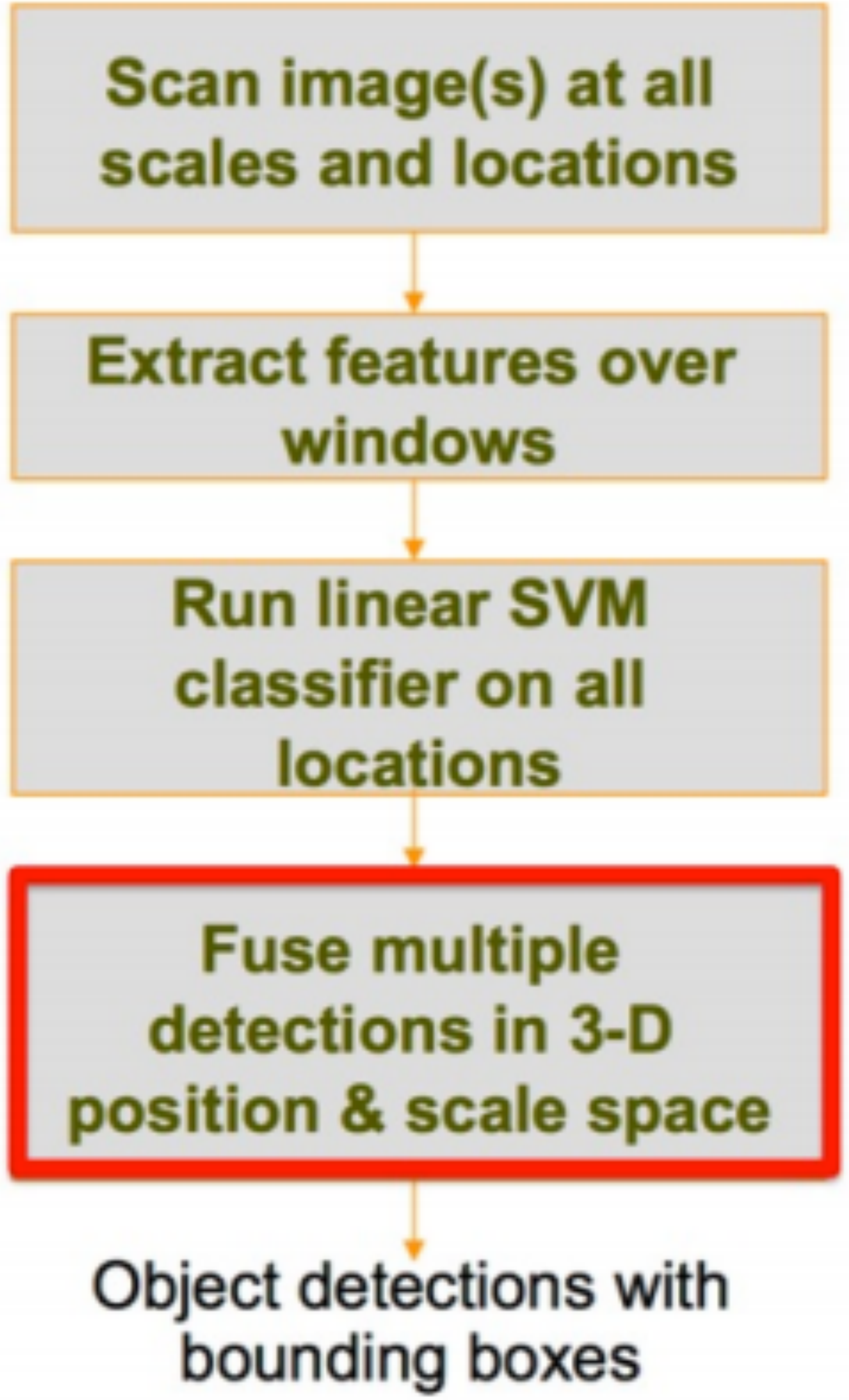
- Computing the score $\mathbf{w}^T \cdot \mathbf{x} + b$ in every location is the same as performing **cross-correlation with template w** (and add b to result).



[Pic from: R. Girshik]

Slide credit: Sanja Fidler

IV. Non-Maxima Suppression (NMS)



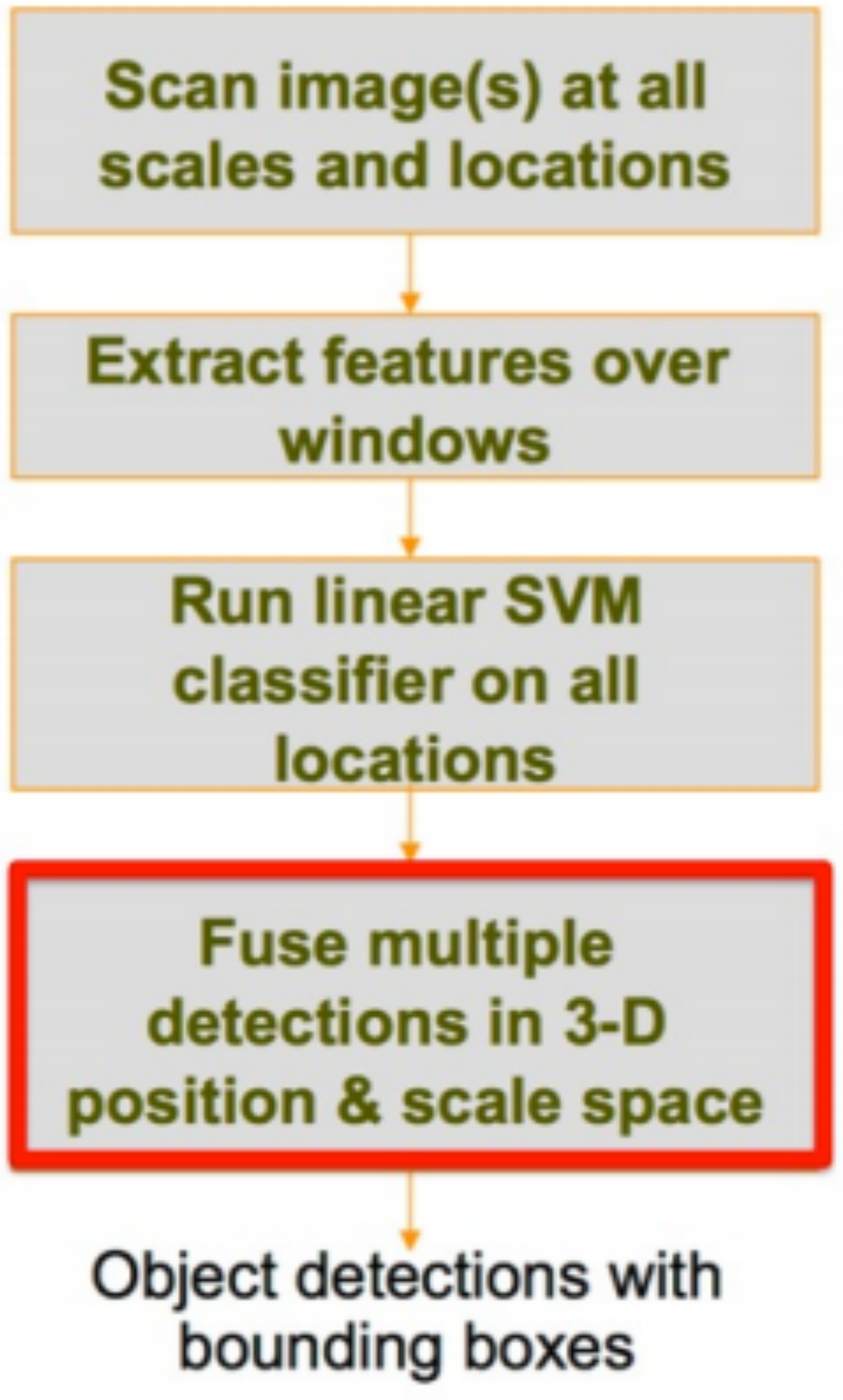
Non-maxima suppression (NMS)

$$\text{overlap} = \frac{\text{area}(box_1 \cup box_2)}{\text{area}(box_1 \cap box_2)} > 0.5 \quad \Rightarrow \quad \boxed{\text{remove } box_2}$$

- Remove all boxes that overlap more than XX (typically 50%) with the chosen box

Slide credit: Sanja Fidler

IV. Non-Maxima Suppression (NMS)



Non-maxima suppression (NMS)

- Greedy algorithm.
- At each iteration pick the highest scoring box.
- Remove all boxes that overlap more than XX (typically 50%) with the chosen box

Slide credit: Sanja Fidler

HOG detector: summary

Dalal & Triggs '05

- Histrogram of Oriented Gradients (HOG)
- SVM training
- Sliding window detection

Slide credit: Sanja Fidler, Ross Girshick

Example 3: How can we deal with this guy?

Dalal & Triggs ’05
- Histrogram of Oriented Gradients (HOG)
- SVM training
- Sliding window detection

Slide credit: Sanja Fidler, Ross Girshick
Pic credit: http://www.deceptology.com/2011/02/participants-in-facebook-game-of-lying.html

HOG detector: limitations

Dalal & Triggs '05
- Histrogram of Oriented Gradients (HOG)
- SVM training
- Sliding window detection

We need flexible models!

Fischler & Elschlager '73
Felzenszwalb & Huttenlocher '00
- Pictorial structures
- Weak appearance models
- Non-discriminative training

Slide credit: Sanja Fidler, Ross Girshick
Pic credit: http://www.deceptology.com/2011/02/participants-in-facebook-game-of-lying.html

The DPM Detector

P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan

Object Detection with Discriminatively Trained Part Based Models

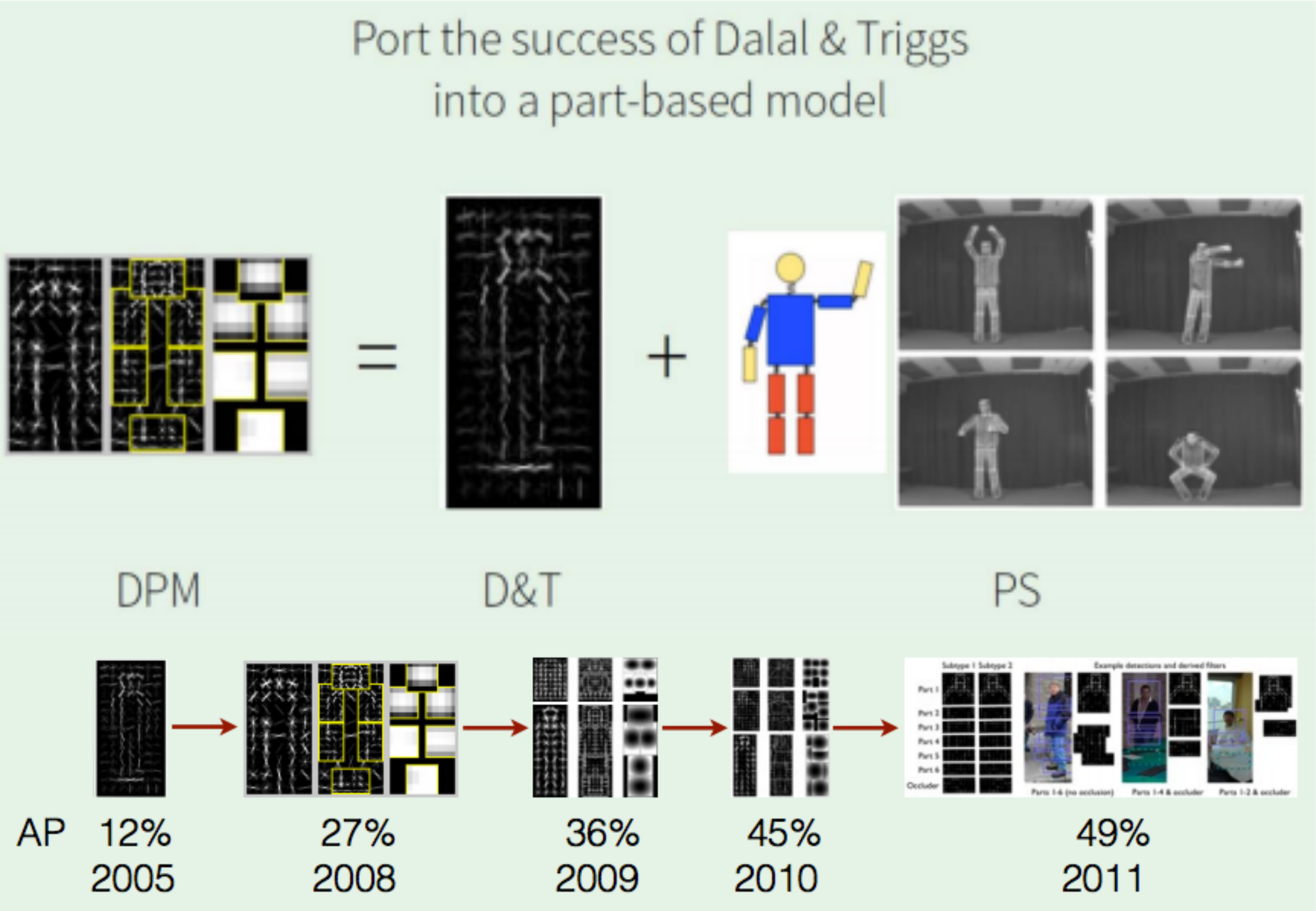
T-PAMI, 2010

Paper: `http://cs.brown.edu/~pff/papers/lsvm-pami.pdf`
Code: `http://www.cs.berkeley.edu/~rbg/latent/`

cited by 5,084

Slide credit: Sanja Fidler

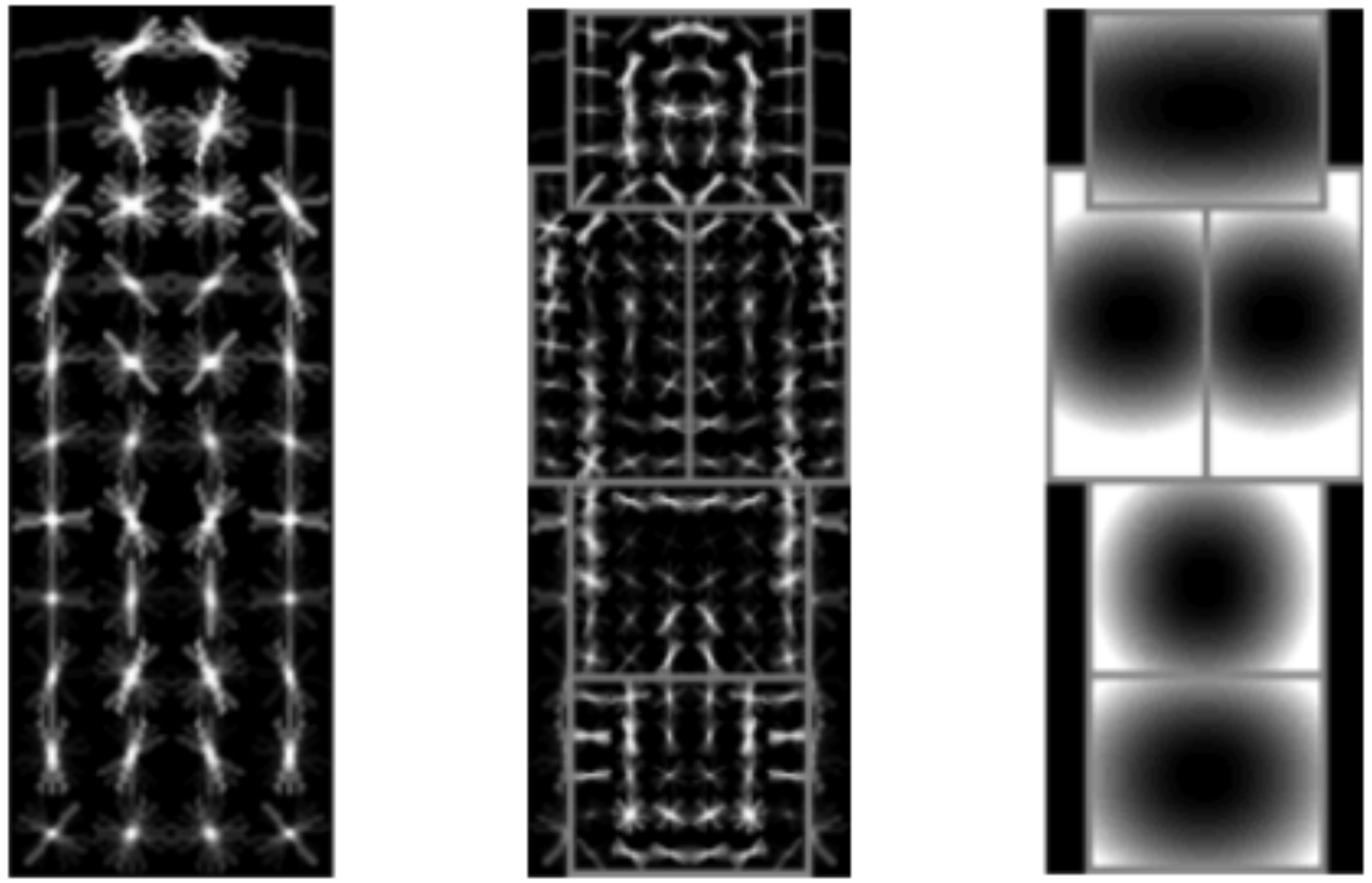
Deformable Part Model (DPM): key idea



Slide credit: Ross Girshick

DPM: Model representation

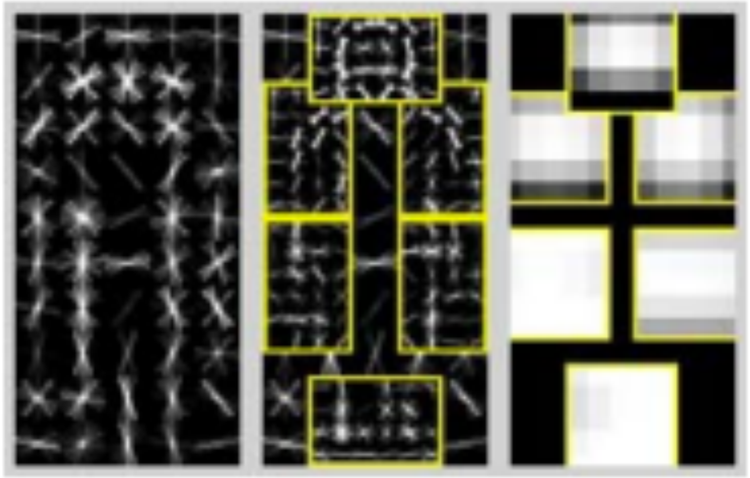
- A model has a root filter F_0 and n part models (F_i, v_i, d_i)
  - F_i: i-th part filter
  - v_i: anchor position of i-th part relative to the root
  - d_i: deformation parameters for i-th part

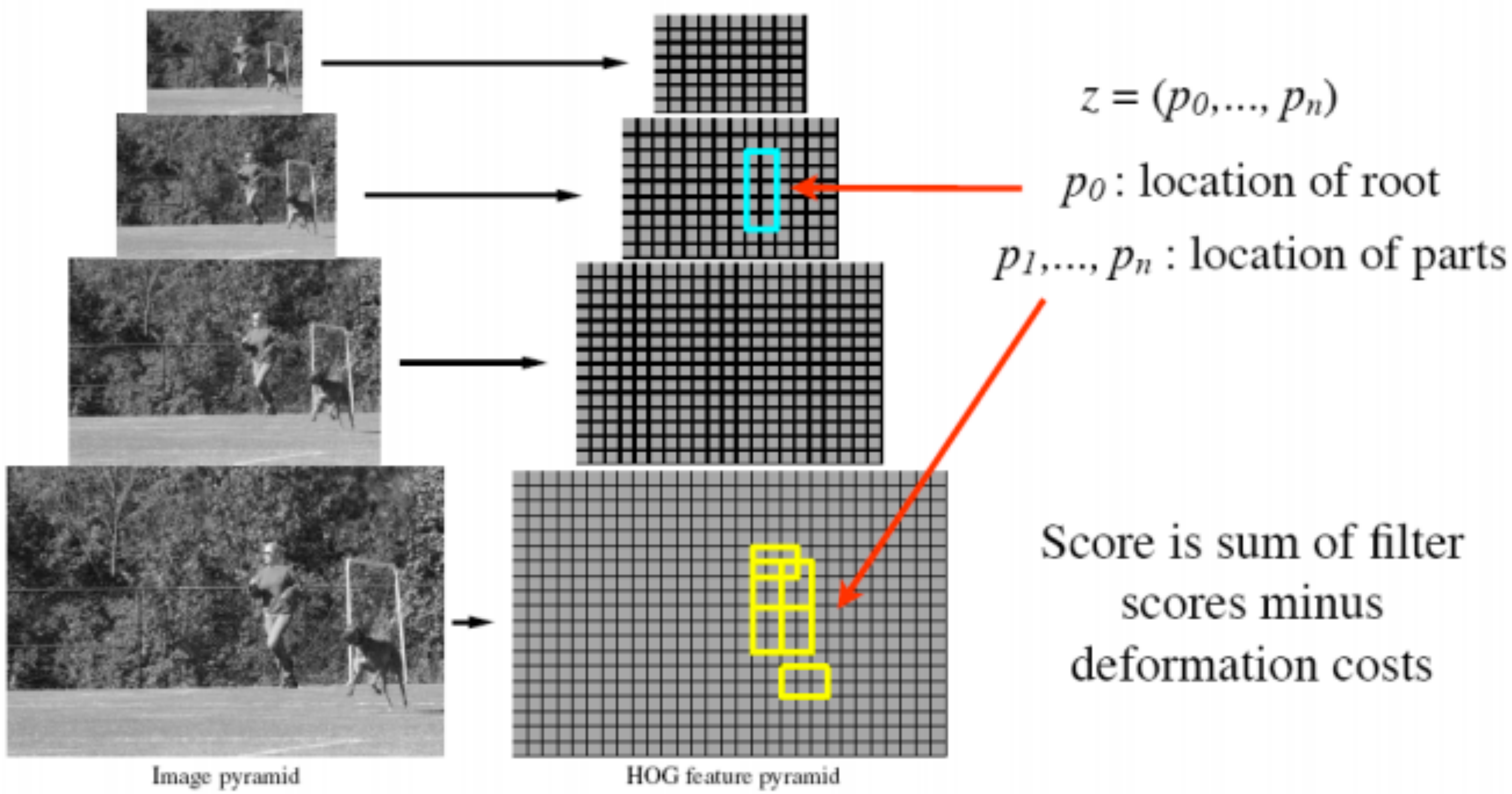


Coarse root filter | Higher resolution part filters | Deformation models

DPM: Object Hypothesis

- In HOG feature pyramid
  - root filter - coarser scale
  - part filters - finer scale





Slide credit: Mubarak Shah

DPM: Score of a Hypothesis

$$score(p_0, \ldots, p_n) = \underbrace{\boxed{\sum_{i=0}^{n} F_i \cdot \phi(H, p_i)}}_{\text{data term}} - \underbrace{\boxed{\sum_{i=1}^{n} d_i \cdot \phi_d(dx_i, dy_i)}}_{\text{spatial prior}} + b$$

Filters

Feature of subwindow at location p_i

Deformation parameters

Displacement of part i relative to its anchor position

Bias

Score of a hypothesis z is

$$score(z) = \beta \cdot \psi(H, z)$$

where

$$\beta = (F_0, \ldots, F_n, d_1, \ldots, d_n, b) \quad \text{Unknown}$$

$$\psi(H, z) = (\phi(H, p_0), \ldots, \phi(H, p_n), -\phi_d(dx_1, dy_1), \ldots, -\phi_d(dx_n, dy_n), 1) \quad \text{Known}$$

DPM: Score of a Hypothesis

Bias

$$score(p_0, \ldots, p_n) = \overbrace{\sum_{i=0}^{n} F_i \cdot \phi(H, p_i)}^{\text{data term}} - \overbrace{\sum_{i=1}^{n} d_i \cdot \phi_d(dx_i, dy_i)}^{\text{spatial prior}} + b$$

Filters

Feature of subwindow at location p_i

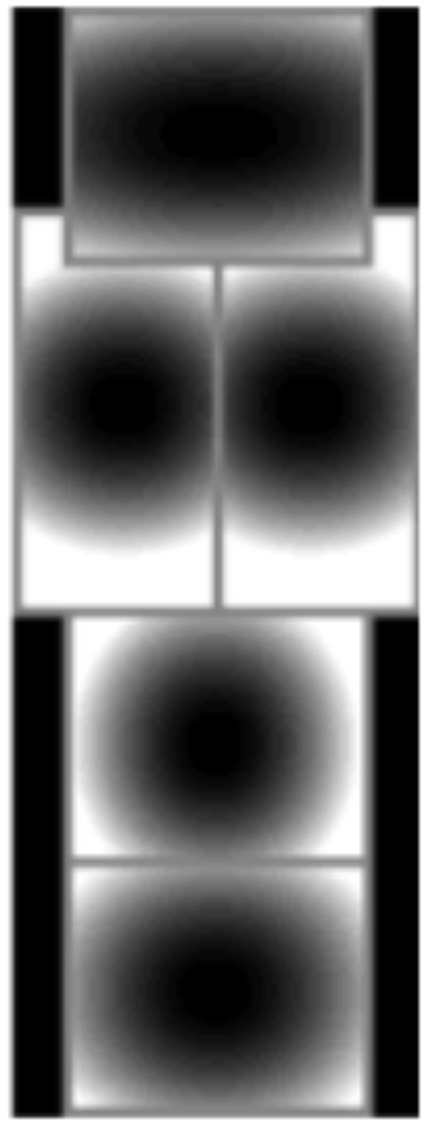
Deformation parameters

Displacement of part i relative to its anchor position

$$\phi_d(dx, dy) = (dx, dy, dx^2, dy^2)$$

$$d_i = (0, 0, 1, 1)$$

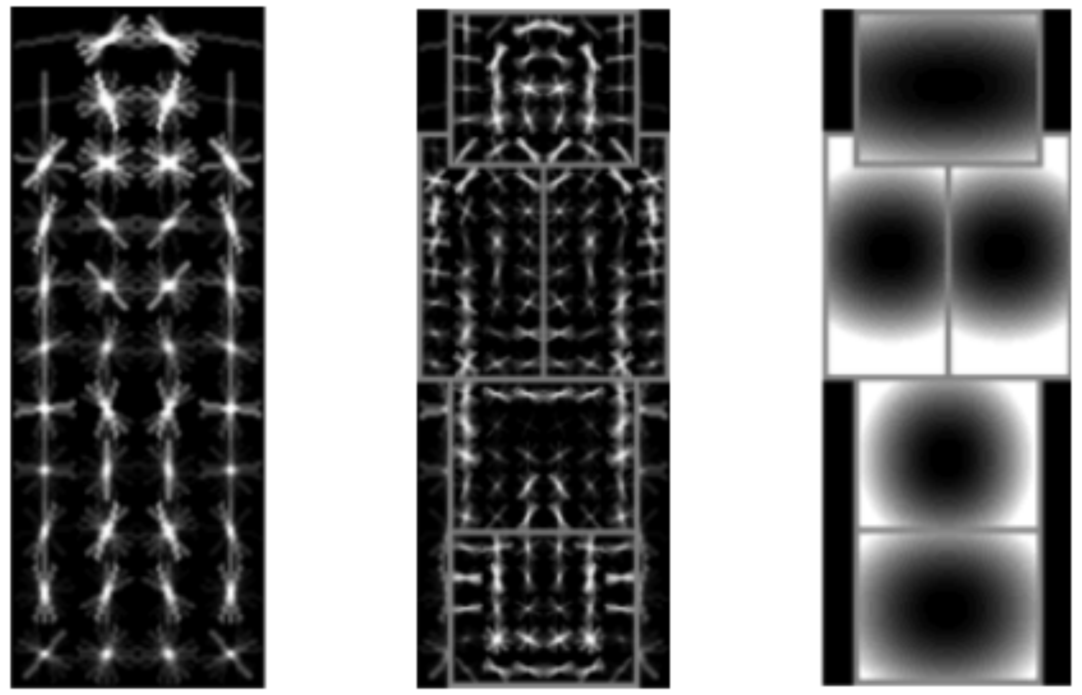
Initial Value
(to be learned)



Deformation models

Slide credit: Mubarak Shah

DPM: Detection



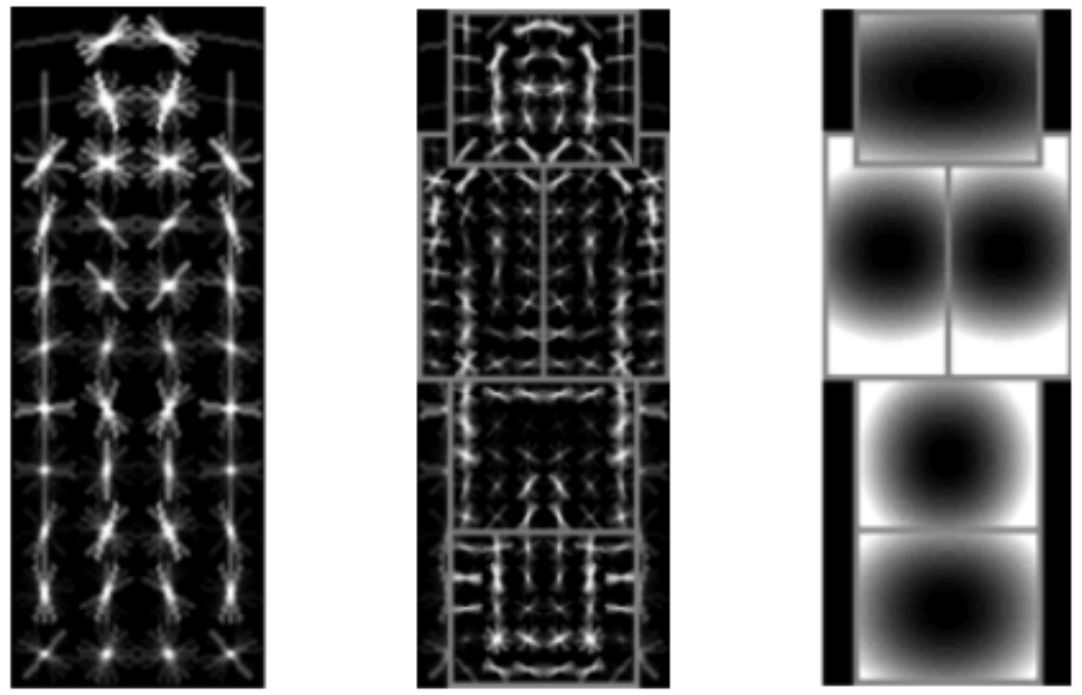
- The overall score of a root location is computed according to the best possible placement of the parts

$$score(p_0) = \max_{p_1,\ldots,p_n} score(p_0, \ldots, p_n)$$

- High-scoring root locations define detections
- Sliding-window approach
- Efficient computation (O(nk)): dynamic programming + generalized distance transforms

Slide credit: Mubarak Shah

DPM: Detection



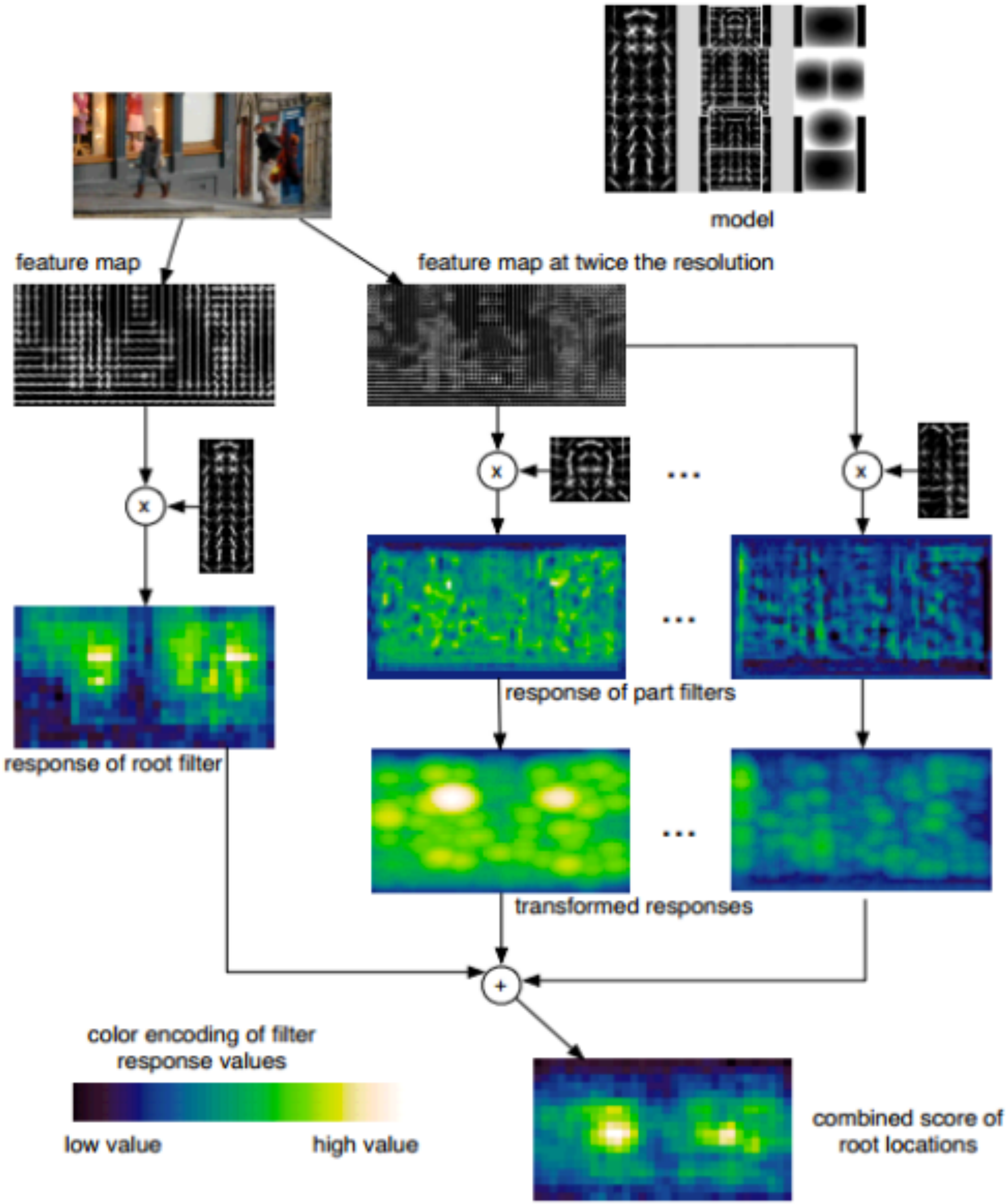
- Distance transform
  - Response of the i-th part filter in the l-th level of the feature pyramid
    $$R_{i,l}(x, y) = F_i \cdot \phi(H, (x, y, l))$$
  - Transformed response, given root is at *(x,y)*
    $$D_{i,l}(x, y) = \max_{dx,dy}(R_{i,l}(x + dx, y + dy) - d_i \cdot \phi_d(dx, dy))$$

$$\phi_d(dx, dy) = (dx, dy, dx^2, dy^2)$$

$$d_i = (0, 0, 1, 1)$$

(x,y)

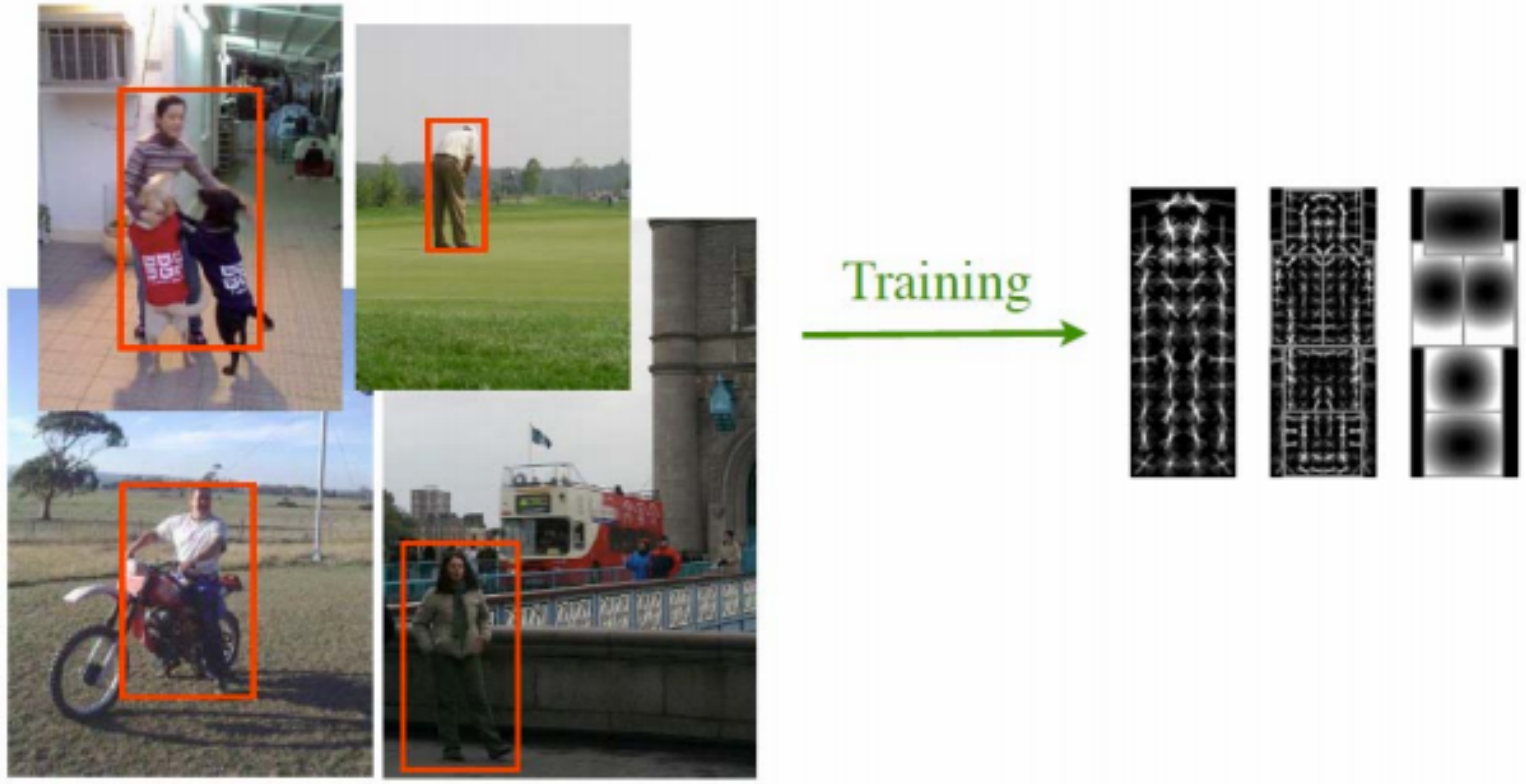
DPM: Detection



Slide credit: Mubarak Shah, Ross Girshick

DPM: Training

- Positive training examples are labeled with bounding boxes
- No part location is available during training (latent)
- Aim: learn model parameters $\beta = (F_0, \dots, F_n, d_1, \dots, d_n, b)$



DPM: Latent Variables

- The positions of the parts are not given in both the training and the testing images
- The variables that exist but not known in training samples are called latent variables
- The learning algorithm must be able to find/discover the optimal values for the latent variables, namely the position of the parts.

Slide credit: Mubarak Shah

DPM: Training

- The classifier scores an example x by

$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z)$$

  - β: the model parameters
  - z: latent values
  - $Z(x)$: the possible latent values for example x

Slide credit: Mubarak Shah

DPM: Training

- Minimize the objective function

$$L_D(\beta) = \frac{1}{2} \parallel \beta \parallel^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i))$$

  - Labeled training examples $D = (\langle x_1, y_1 \rangle, \ldots, \langle x_n, y_n \rangle)$
  - $y_i \in \{-1, 1\}$

Slide credit: Mubarak Shah

DPM: Latent SVM

- A latent SVM is semi-convex
  - $f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z)$ is convex in β
  - For negative examples ($y_i = -1$), the hinge loss is convex
    $$\max\left(0, 1 - y_i f_\beta(x_i)\right) = \max(0, 1 + f_\beta(x_i))$$
    (the maximum of two convex function)
  - For positive examples ($y_i = 1$), the hinge loss is not convex
    $$\max\left(0, 1 - y_i f_\beta(x_i)\right) = \max(0, 1 - f_\beta(x_i))$$
    (the maximum of a convex function and a concave function)
  - If the latent value for positive examples are fixed, the hinge loss is convex

DPM: Latent SVM

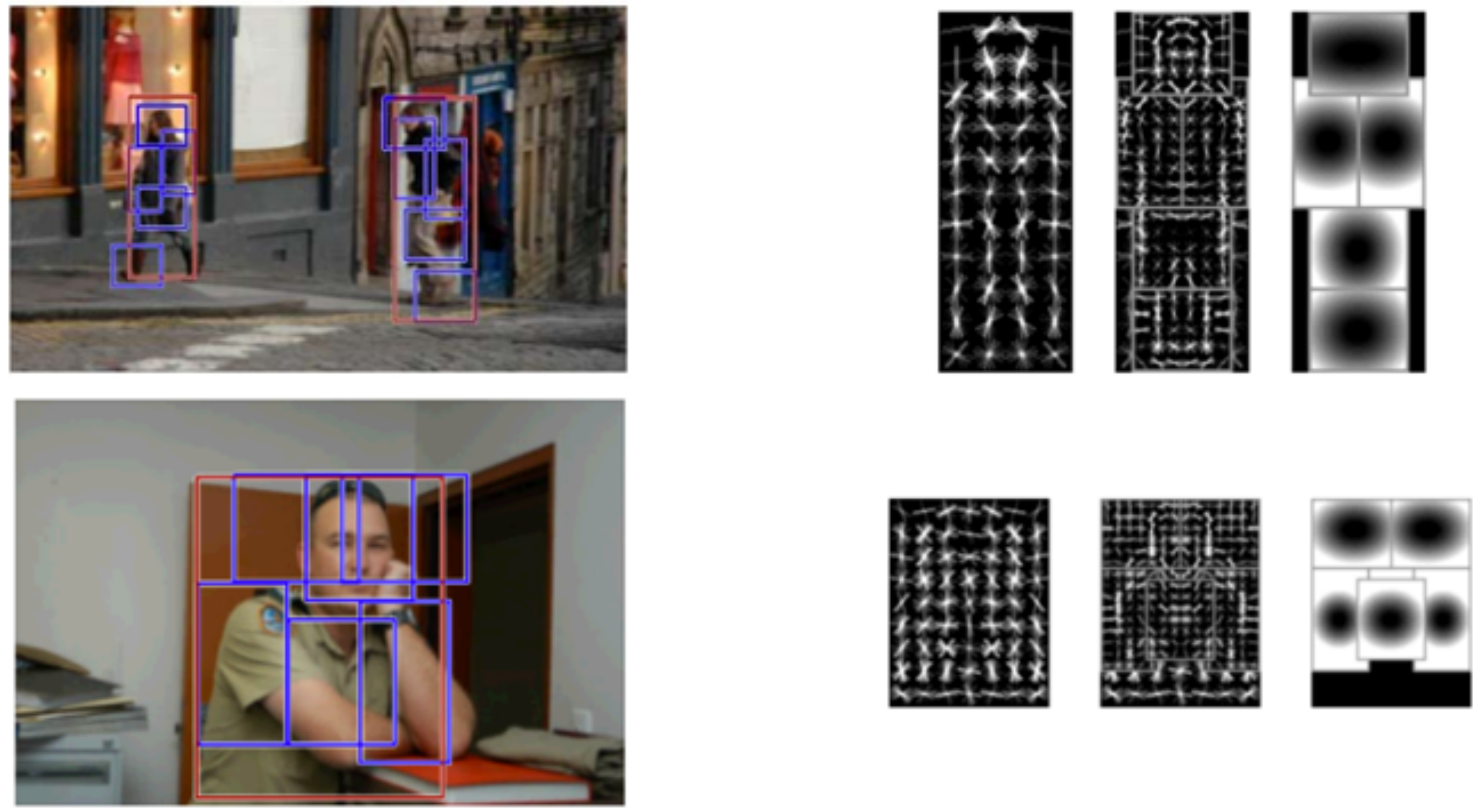
- Initialize β using standard SVM by assuming the same parts locations for all the positive examples
- Iterative optimization:
  - Relabel positive examples: fix β, find the best z for each positive example (exactly the same with detection!)
  - Optimize β: fix z, optimize β by solving the convex problem

Slide credit: Mubarak Shah

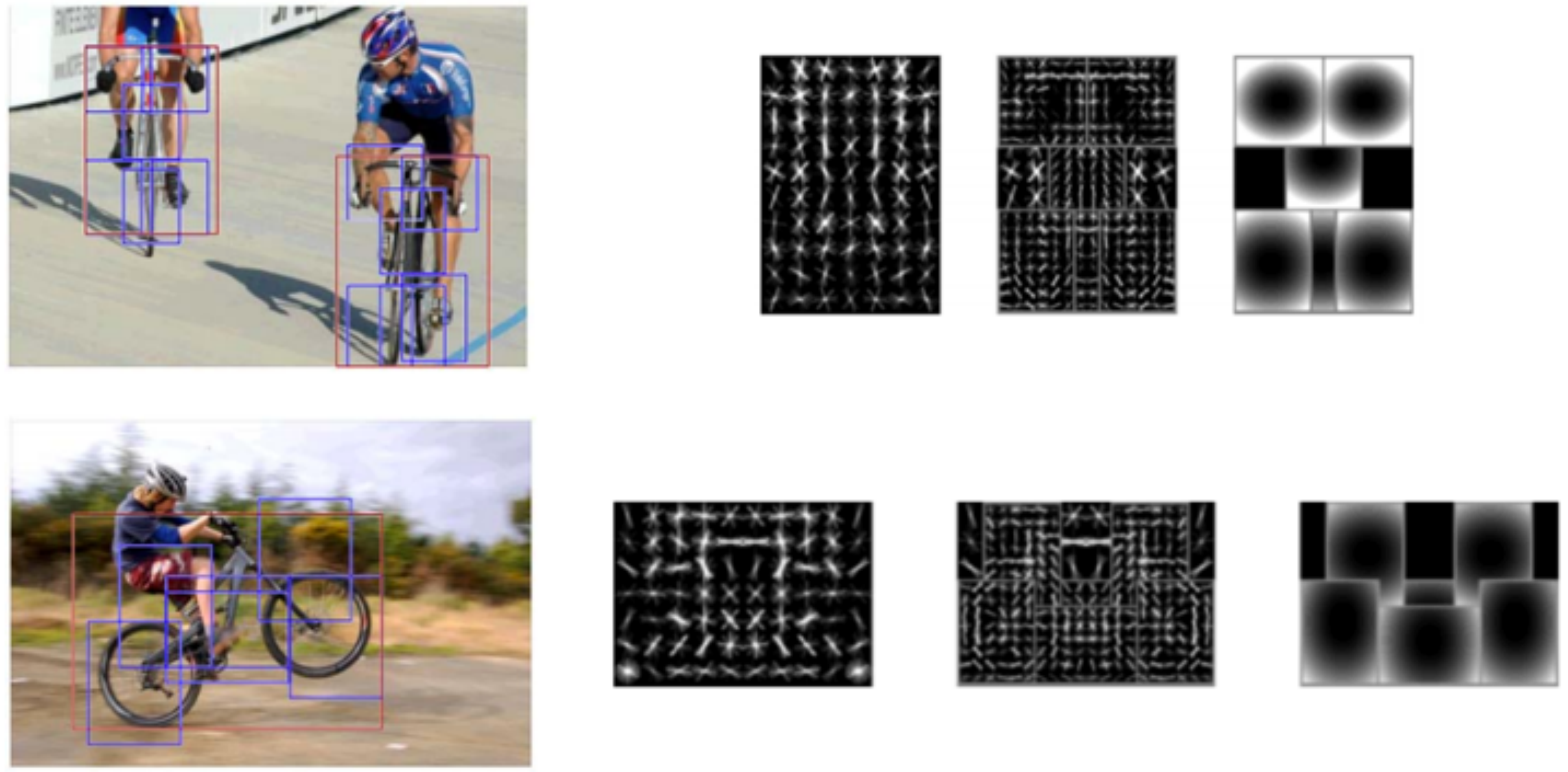
DPM: Mixture model

- A mixture model consists of m components
- Captures extreme intra-class variation
- Split the positive bounding boxes into m groups by aspect ratio

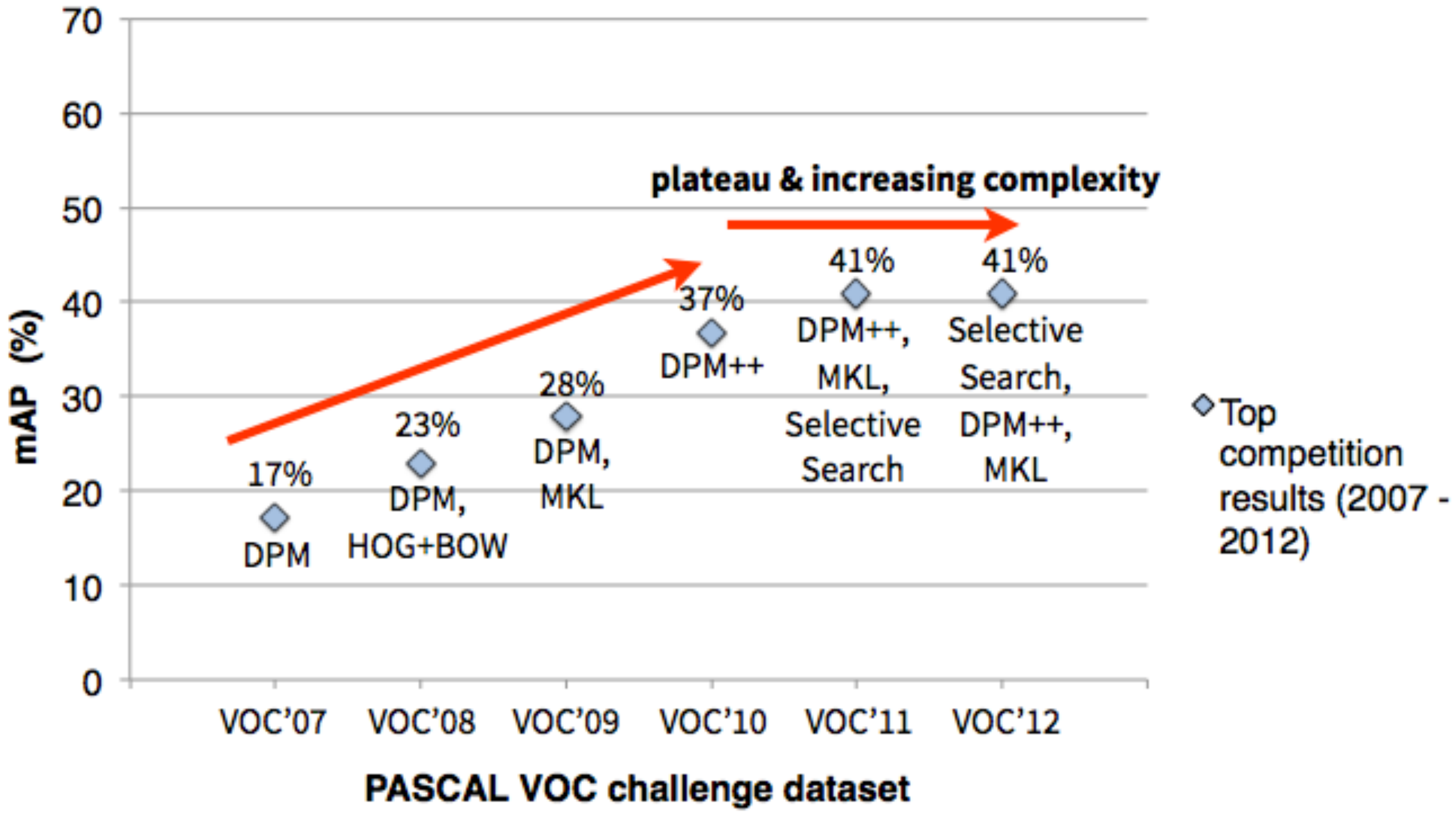
Mixture Model Example - Person



Mixture Model Example - Bicycle



DPM on PASCAL VOC



[Source: http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc20{07,08,09,10,11,12}/results/index.html]

Ross Girshick

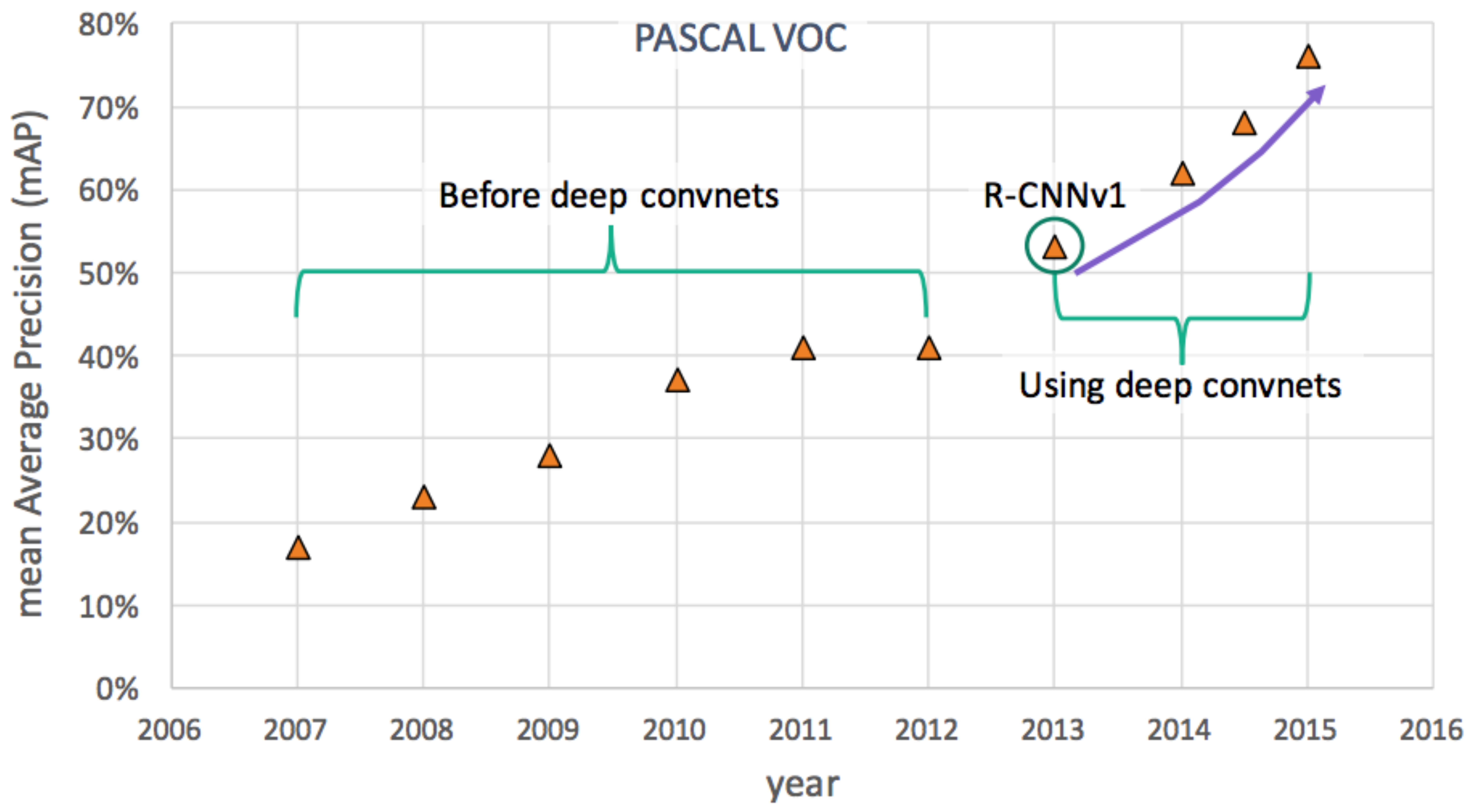
Lifetime Achievement Award
by PASCAL VOC

Slide credit: Ross Girshick

Object detection

- Introduction
- Pre-CNN time
    - HOG detector
    - Deformable Part-based Model
- CNN time
    - Region CNN
    - Fast versions of RCNN
    - YOLO/SSD
- 3D object detection
- Devil's in the details

Object detection renaissance (2013-present)



Slide credit: Renjie Liao

Deep object detection

Object Detection

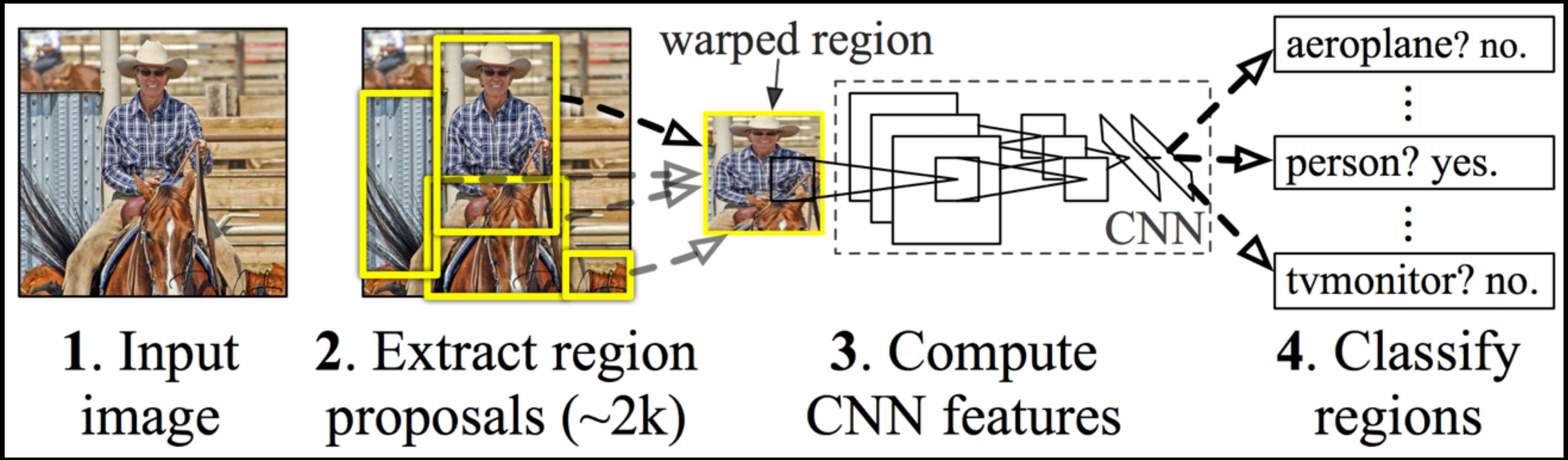
Published: 09 Oct 2015   Category: deep_learning

Jump to...

- Leaderboard
- Papers
  - R-CNN
  - MultiBox
  - SPP-Net
  - DeepID-Net
  - NoC
  - Fast R-CNN
  - DeepBox
  - MR-CNN
  - Faster R-CNN
  - YOLO
  - AttentionNet
  - DenseBox
  - SSD
  - Inside-Outside Net (ION)
  - G-CNN
  - HyperNet
  - MultiPathNet
  - CRAFT
  - OHEM
  - R-FCN
  - MS-CNN
  - PVANET
  - GBD-Net
  - StuffNet
  - Feature Pyramid Network (FPN)
  - YOLOv2
  - DSSD

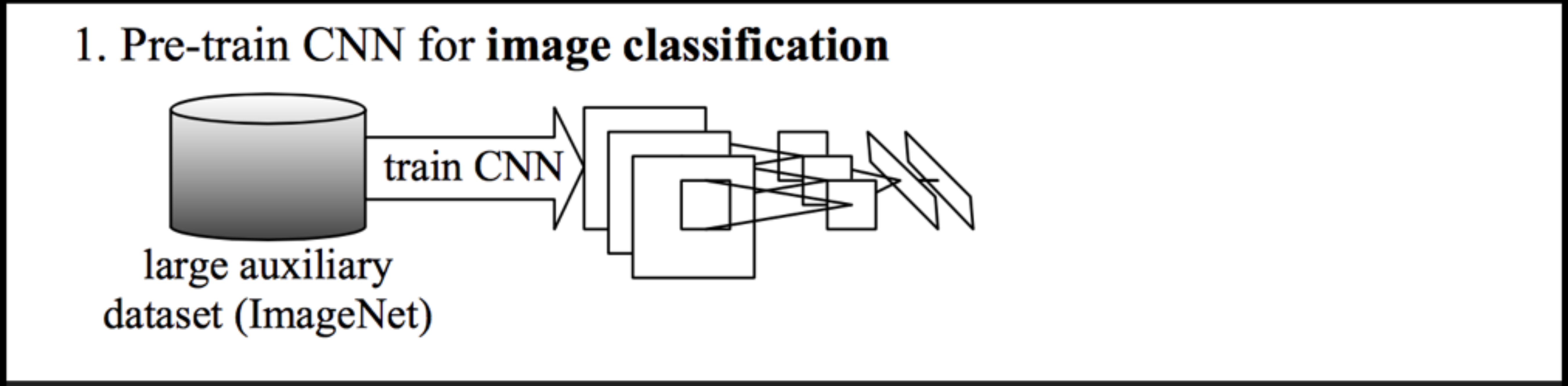
Slide credit: https://handong1587.github.io/deep_learning/2015/10/09/object-detection.html

R-CNN: Regions with CNN features



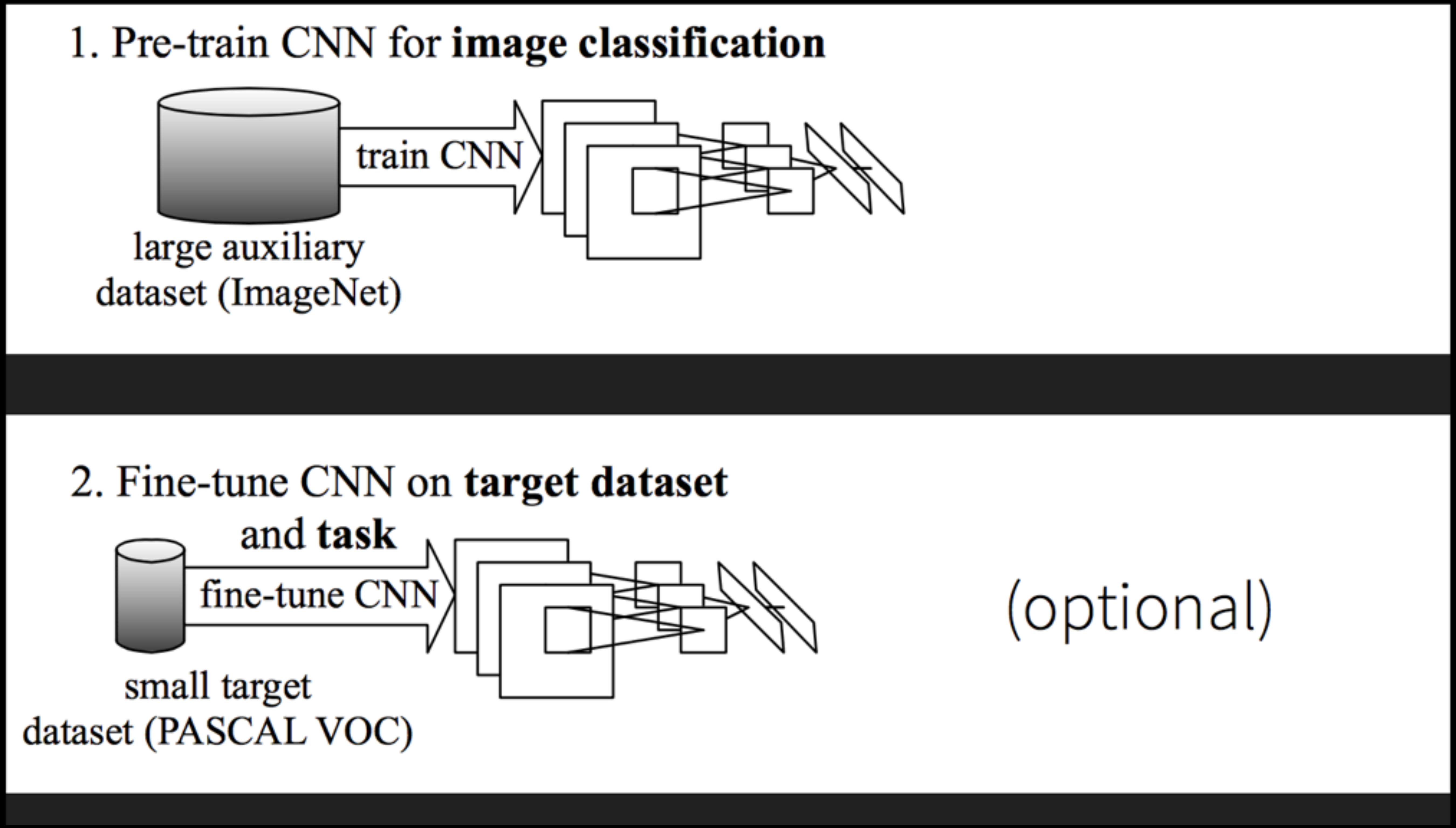
Slide credit: Girshick R, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. CVPR2014

Training



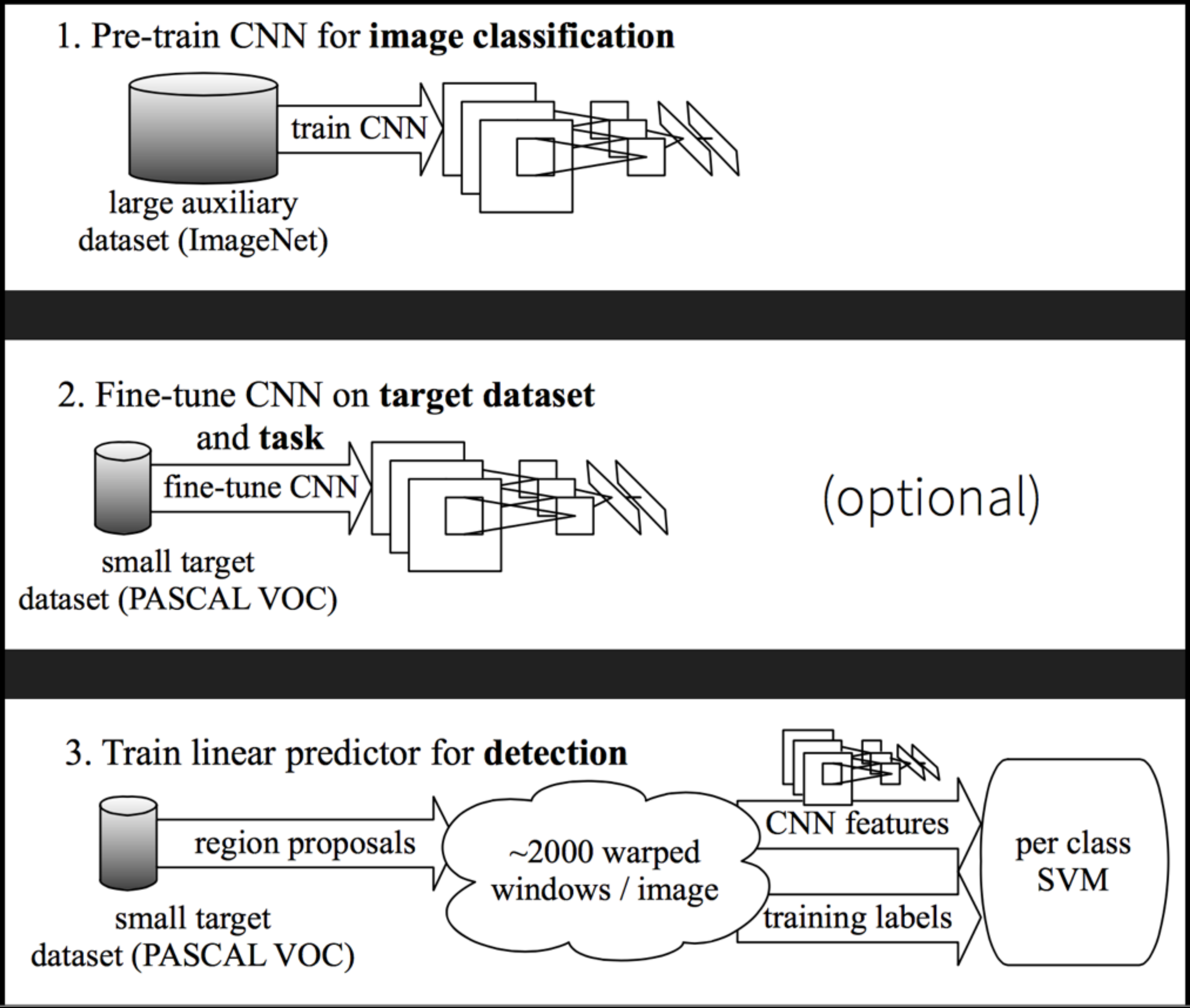
Slide credit: Ross Girshick

Training



Slide credit: Ross Girshick

Training



Slide credit: Ross Girshick

R-CNN Results

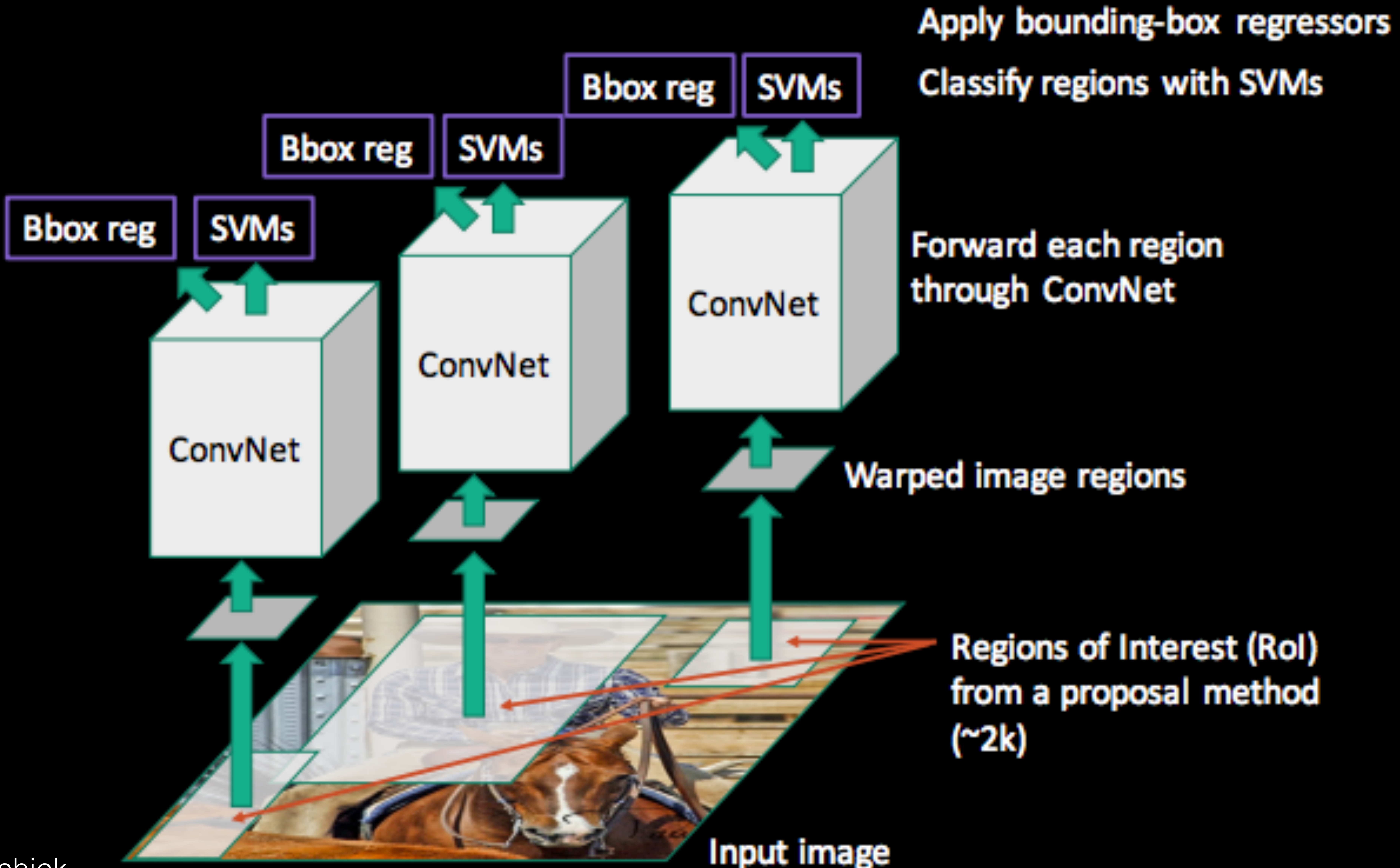
	VOC2007
DPM v5 (Girshick et al. 2011)	33.7%
Regionlets (Wang et al. 2013)	41.7%
R-CNN (AlexNet)	54.2%
R-CNN (AlexNet) + BB	58.5%
R-CNN (VGGNet)	62.2%
R-CNN (VGGNet) + BB	66.0%

R-CNN Results

	VOC2007
DPM v5 (Girshick et al. 2011)	33.7%
Regionlets (Wang et al. 2013)	41.7%
R-CNN (AlexNet)	54.2%
R-CNN (AlexNet) + BB	58.5%
R-CNN (VGGNet)	62.2%
R-CNN (VGGNet) + BB	66.0%

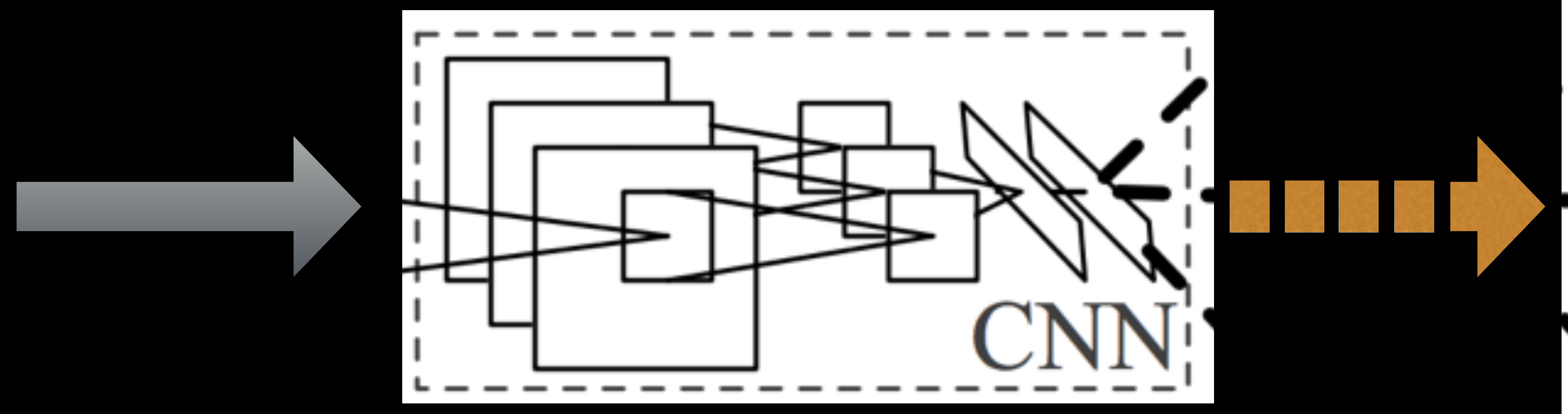
R-CNN (VGGNet)	Time
Train	84 hours
Test	47 s/im

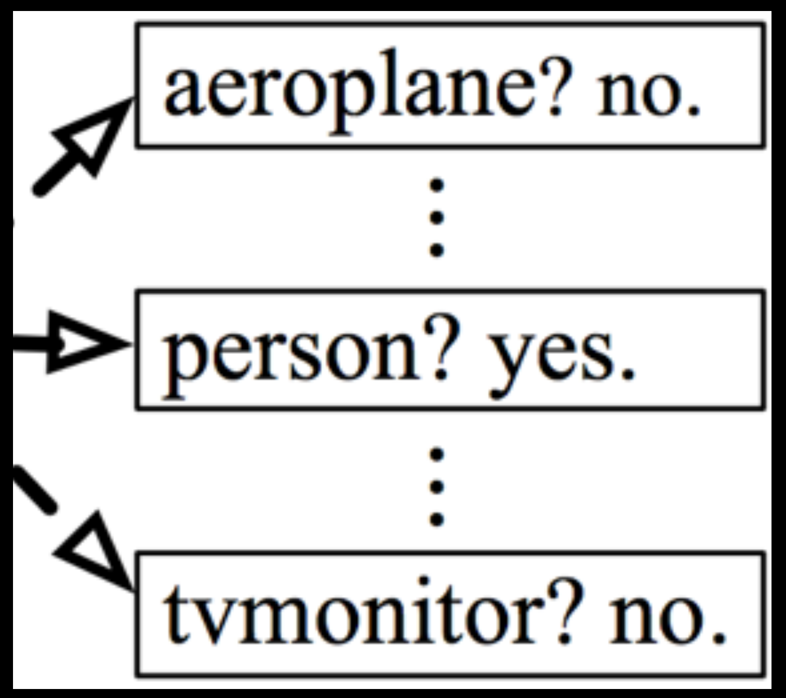
Slow R-CNN



Slide credit: Ross Girshick

Object Detection System



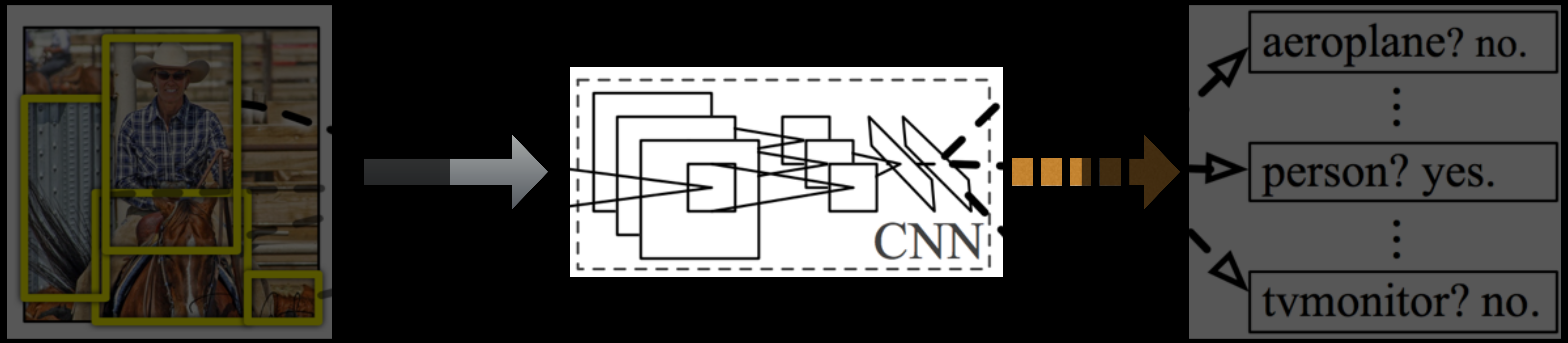


Getting Proposals

Feature Extraction

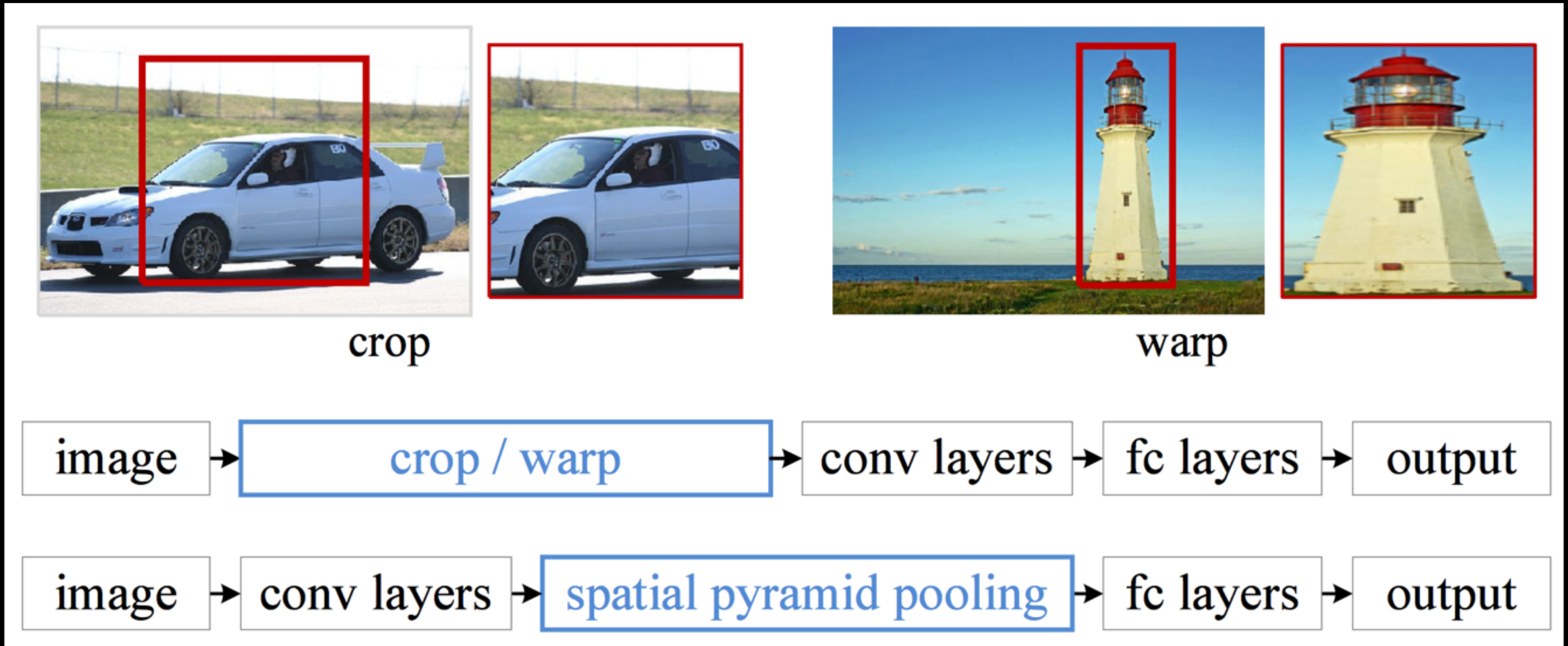
Classifier

Object Detection System



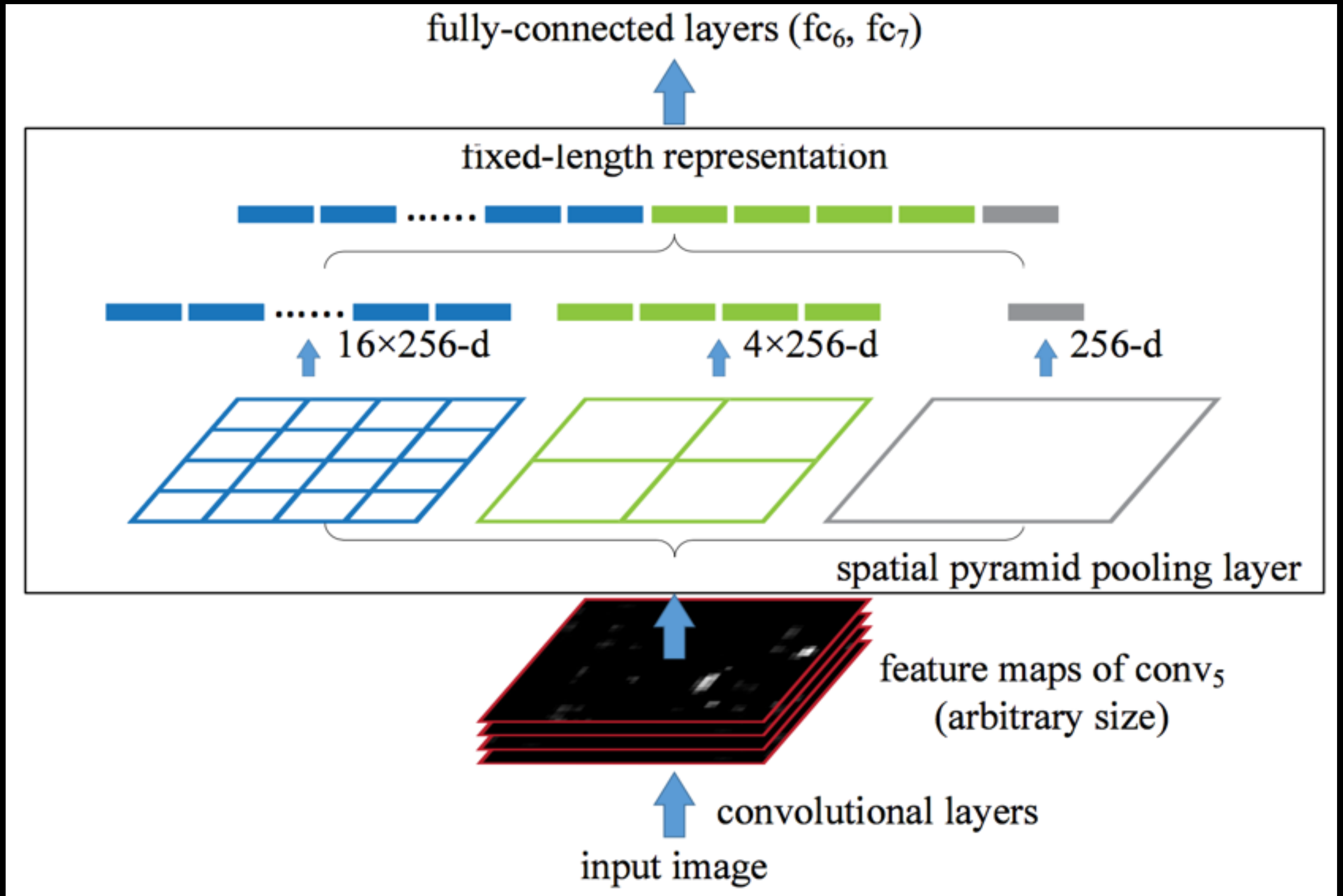
Getting Proposals **Feature Extraction** Classifier

Spatial Pyramid Pooling



Slide credit: K. He, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. ECCV2014

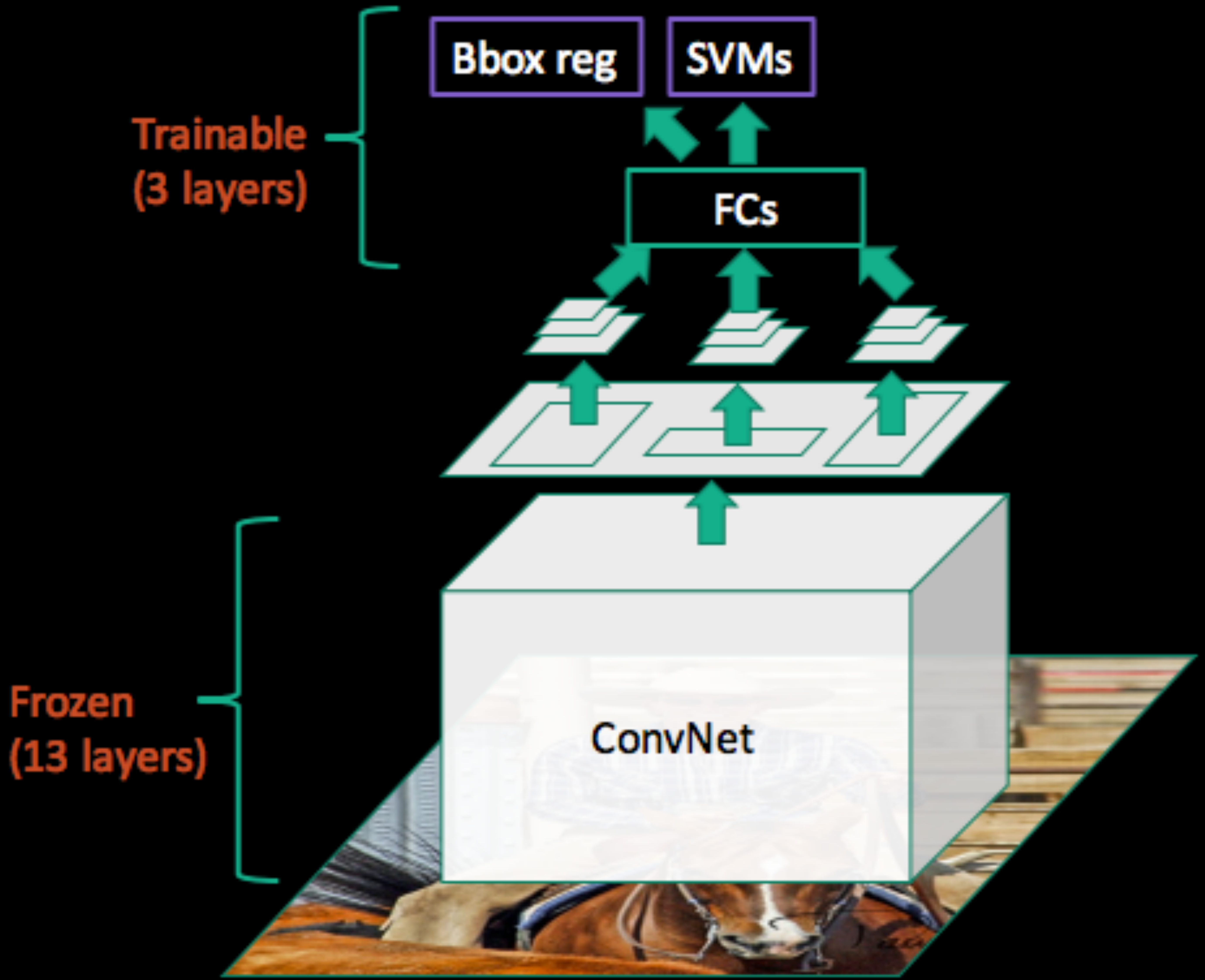
Spatial Pyramid Pooling




```
[pool3x3]
type=pool
pool=max
inputs=conv5
sizeX=5
stride=4

[pool2x2]
type=pool
pool=max
inputs=conv5
sizeX=7
stride=6

[pool1x1]
type=pool
pool=max
inputs=conv5
sizeX=13
stride=13

[fc6]
type=fc
outputs=4096
inputs=pool3x3,pool2x2,pool1x1
```

Slide credit: K. He, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. ECCV2014

SPP-net

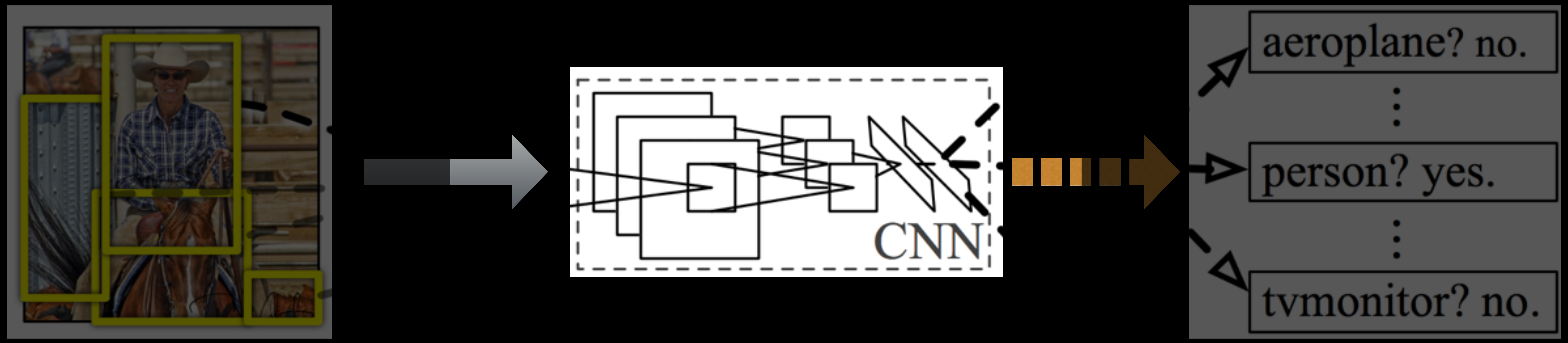


Slide credit: Ross Girshick

SPP-net Results

	VOC2007	Speed
R-CNN (ZFNet)	59.2%	14.5 s/im
R-CNN (VGGNet)	66.0%	47.0 s/im
SPP (ZFNet)	59.2%	0.38 s/im
SPP (VGGNet)	63.1%	2.3 s/im

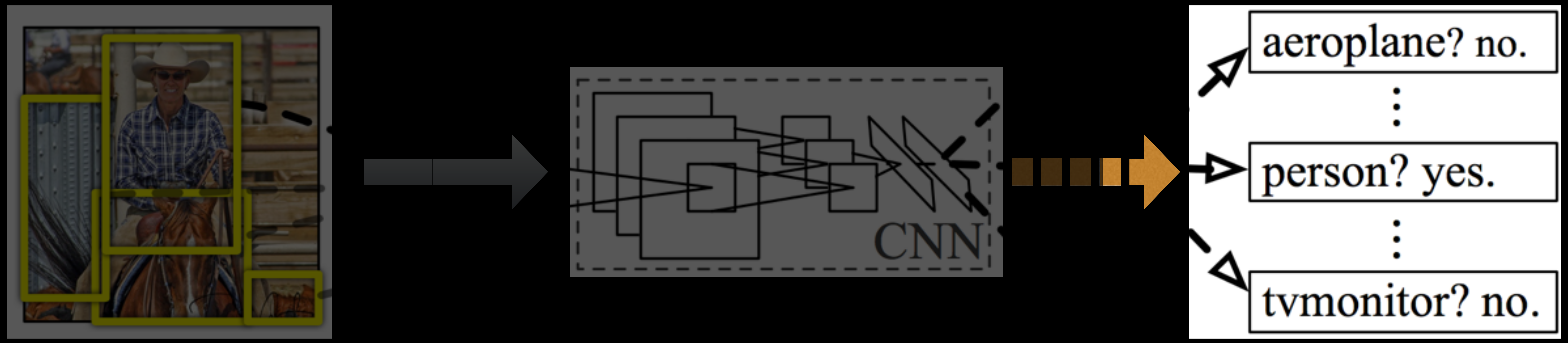
Object Detection System



Getting Proposals **Feature Extraction** Classifier

SPP

Object Detection System



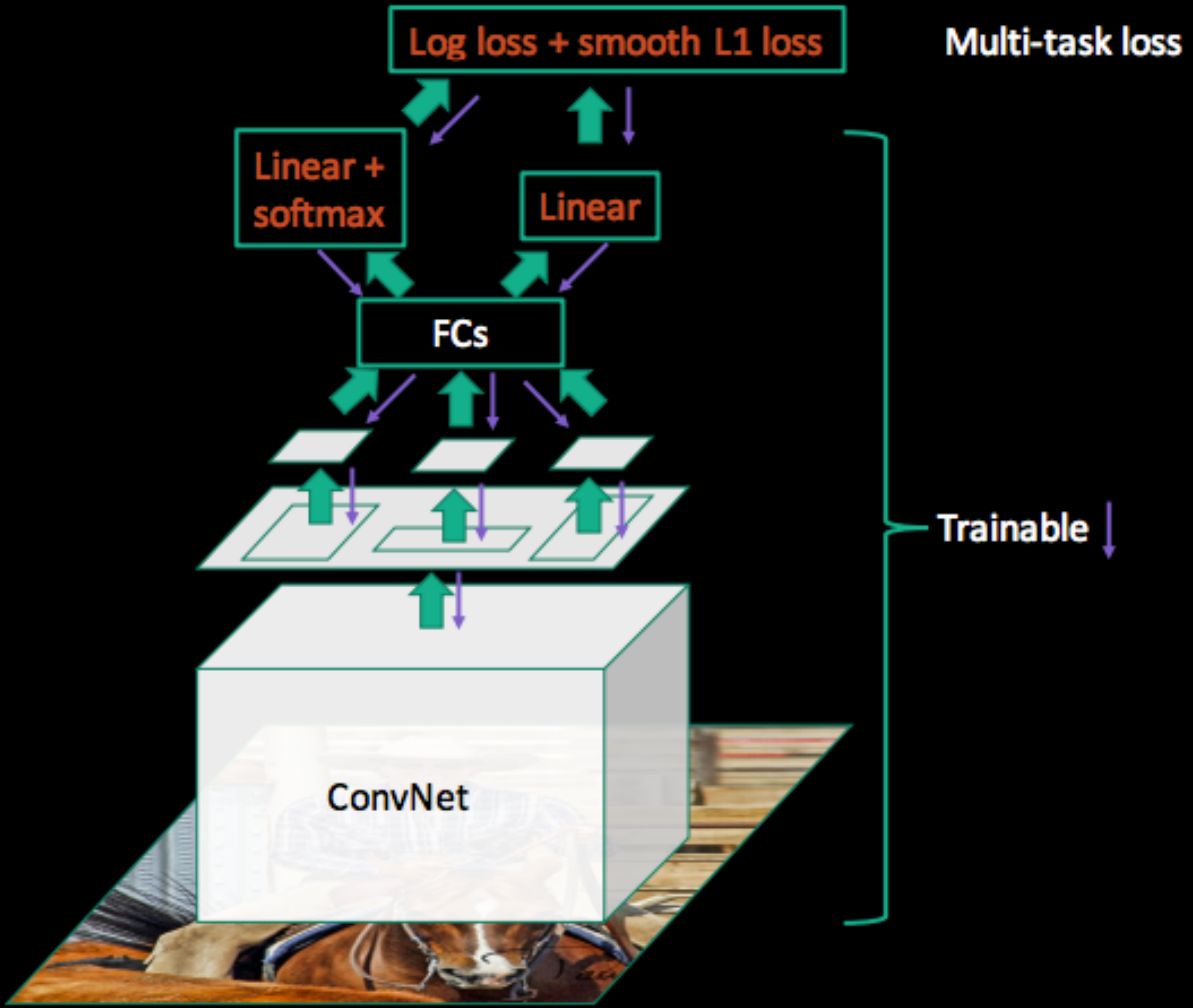
Getting Proposals

Feature Extraction

Classifier

Fast R-CNN

Totally end-to-end!

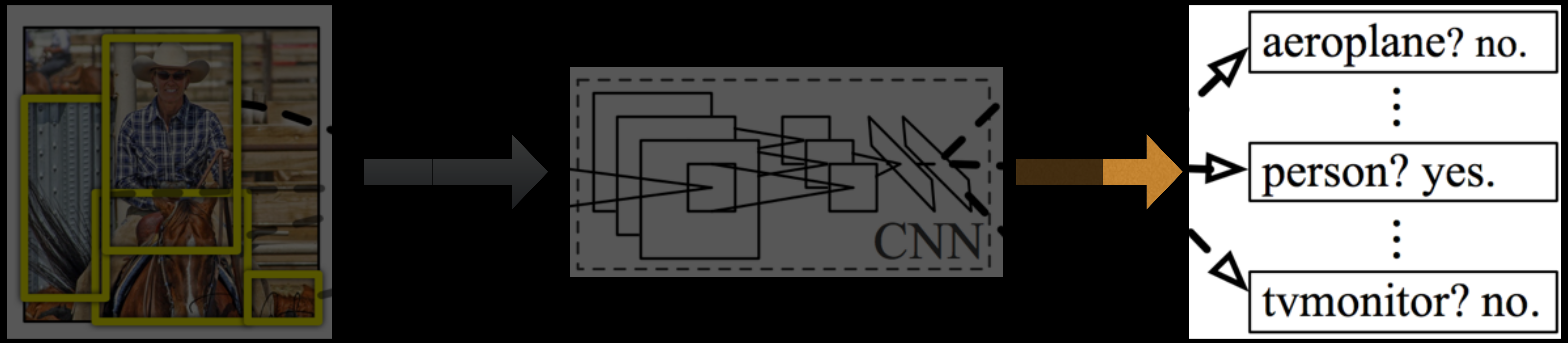


Slide credit: Ross Girshick

Fast R-CNN Results

	VOC2007
SPPNet BB	63.1%
R-CNN BB	66.0%
Fast RCNN	66.9%
Fast RCNN (07+12)	70.0%

Object Detection System



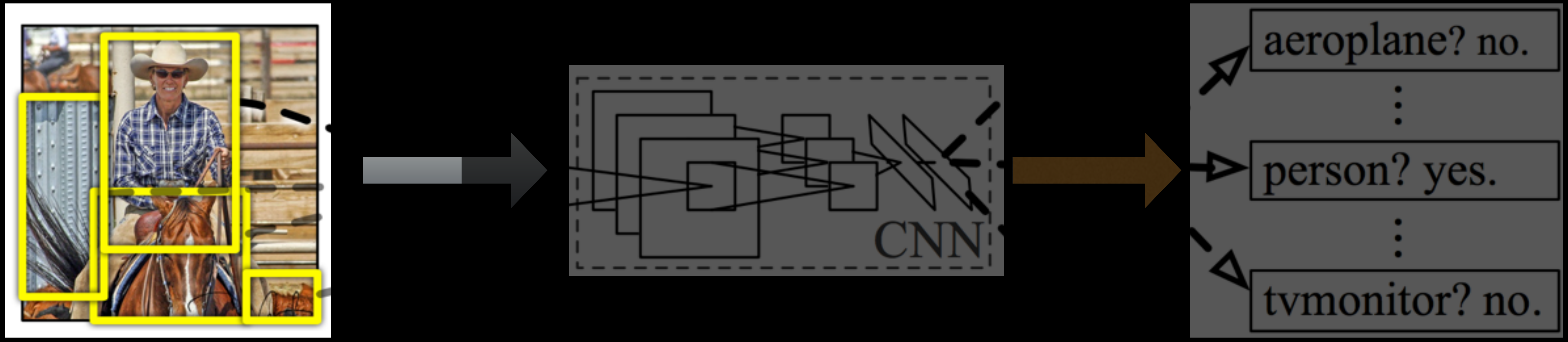
Getting Proposals

Feature Extraction

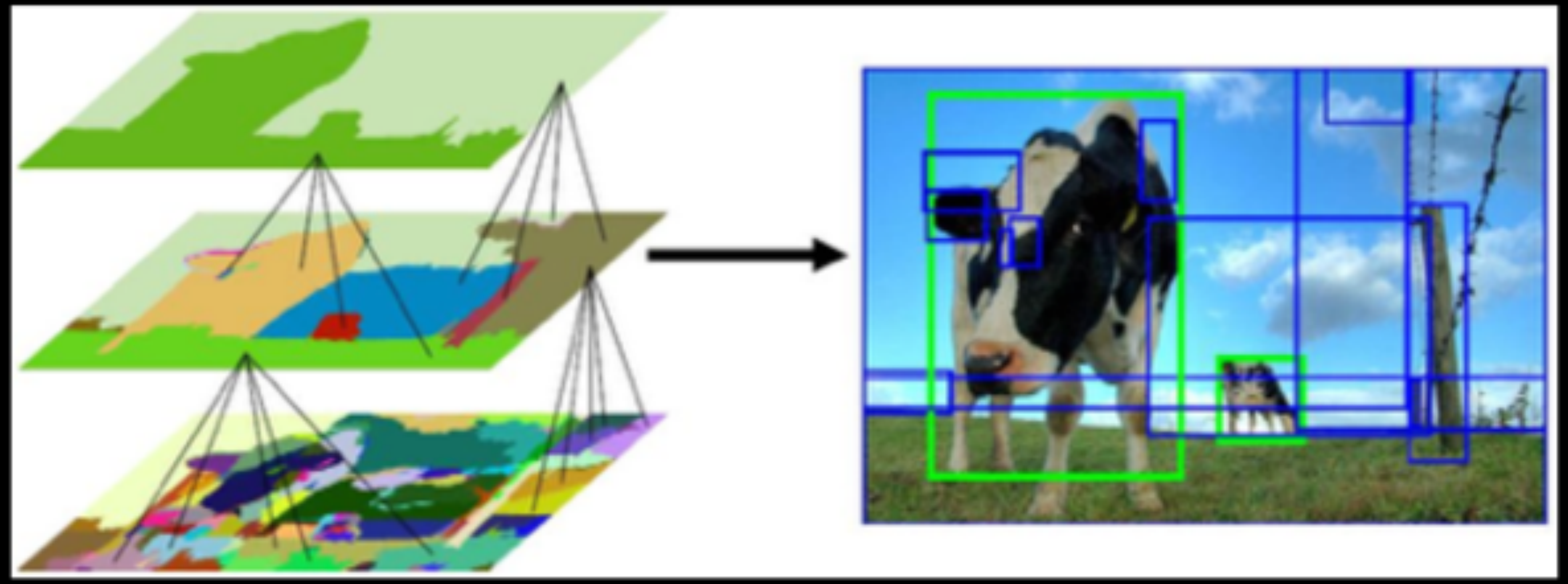
Classifier

Fast R-CNN

Object Detection System

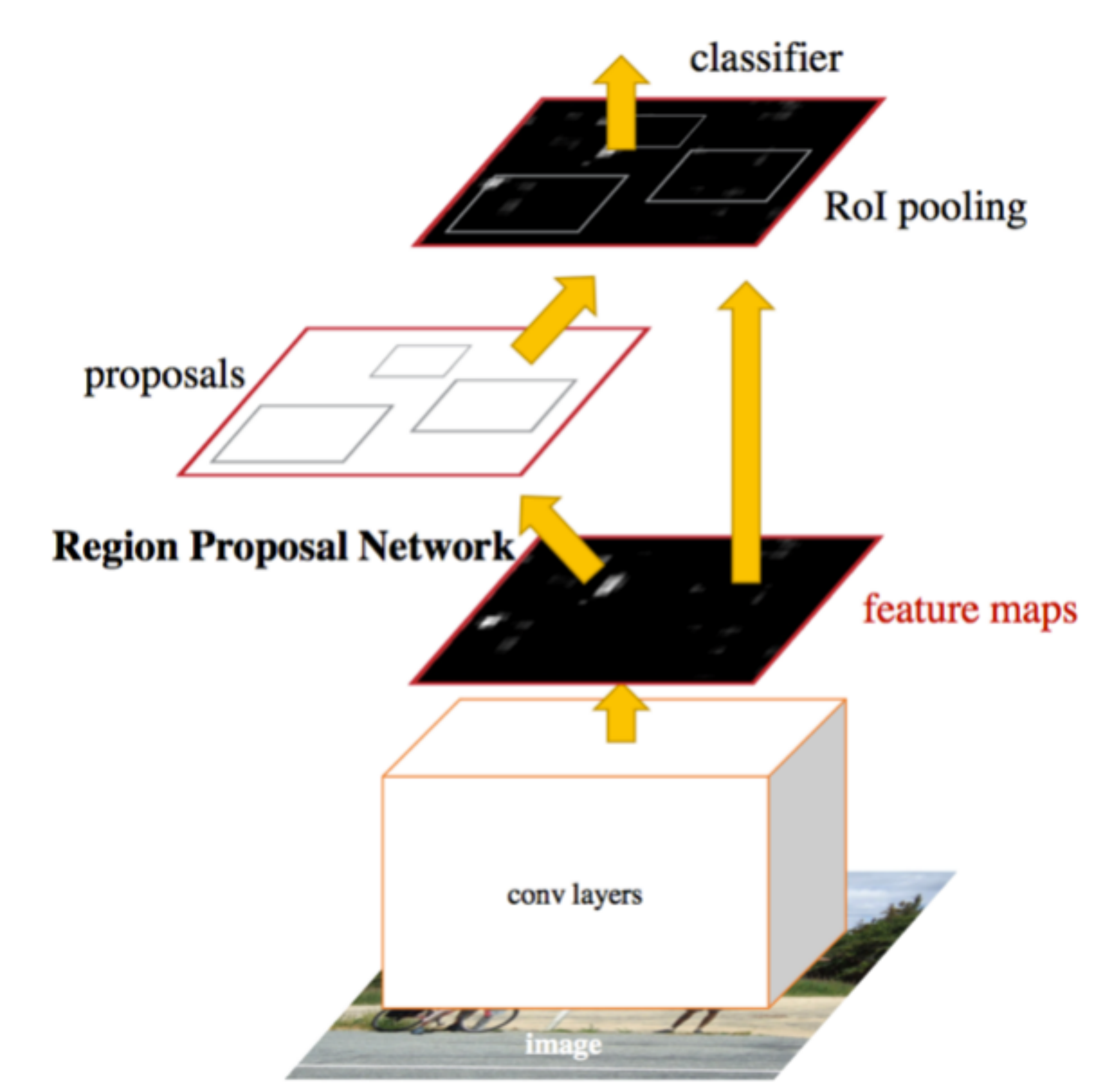


Getting Proposals Feature Extraction Classifier



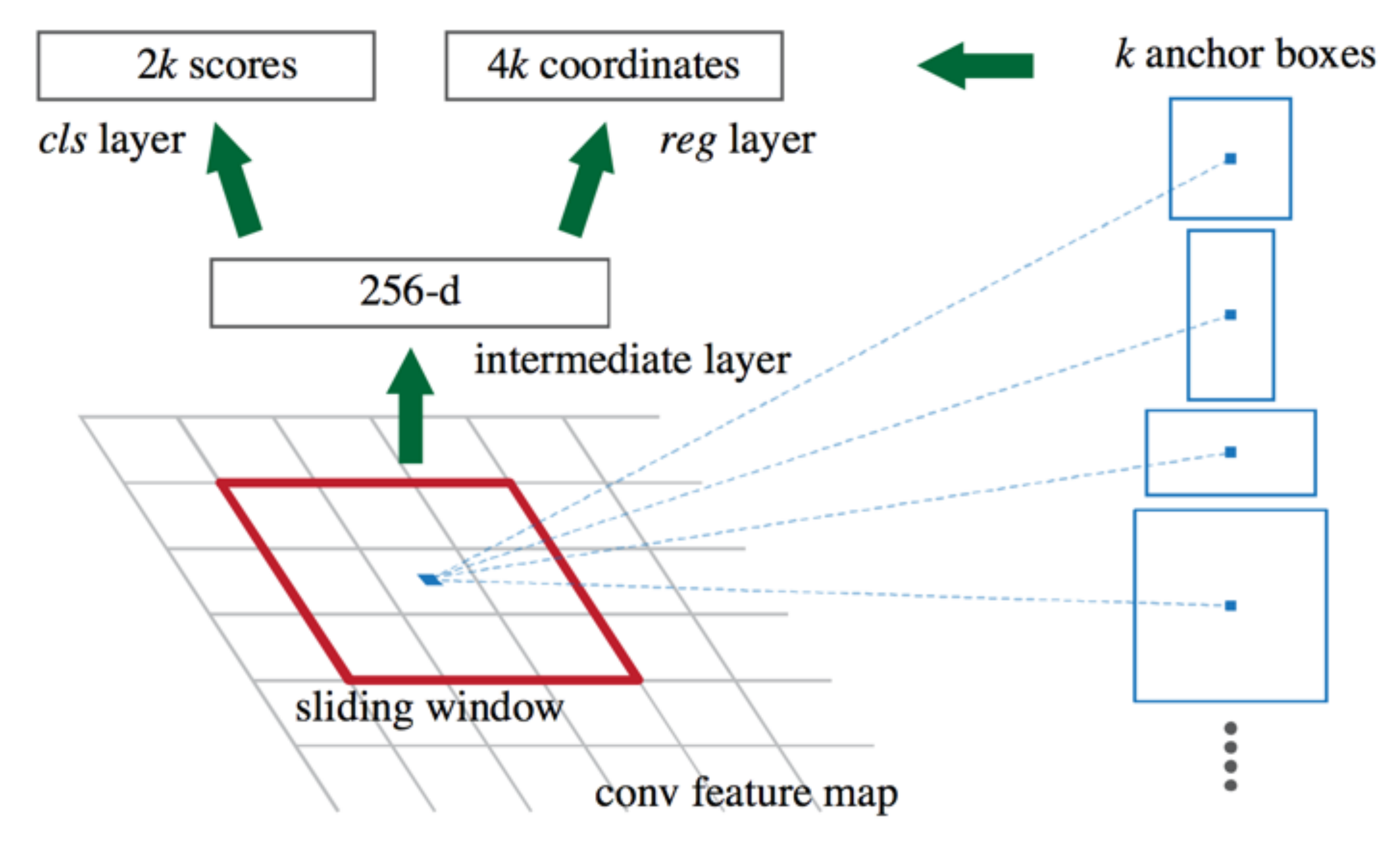
(e.g. selective search)

Faster R-CNN



Slide credit: Ren S, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. NIPS2015

Region Proposal Network



- Sliding window style
- Multi-scale predictions on fix-sized window for efficiency (take advantage of the large receptive field of CNN features)
- Same loss as R-CNN (cls+bbox)

anchor	128^2, 2:1	128^2, 1:1	128^2, 1:2	256^2, 2:1	256^2, 1:1	256^2, 1:2	512^2, 2:1	512^2, 1:1	512^2, 1:2
proposal	188×111	113×114	70×92	416×229	261×284	174×332	768×437	499×501	355×715

Region Proposal Network

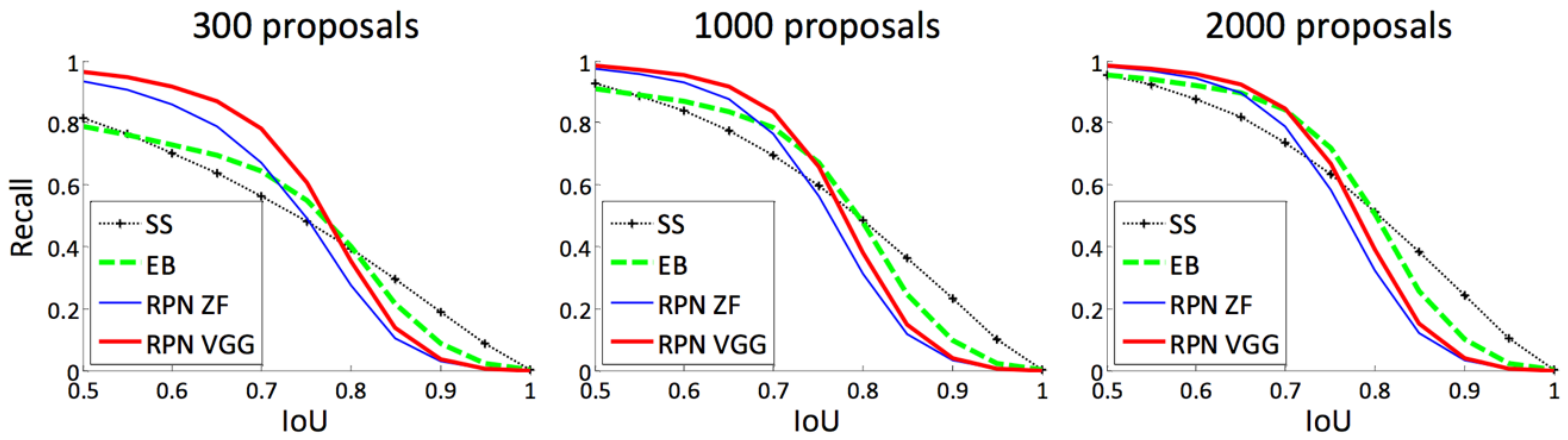


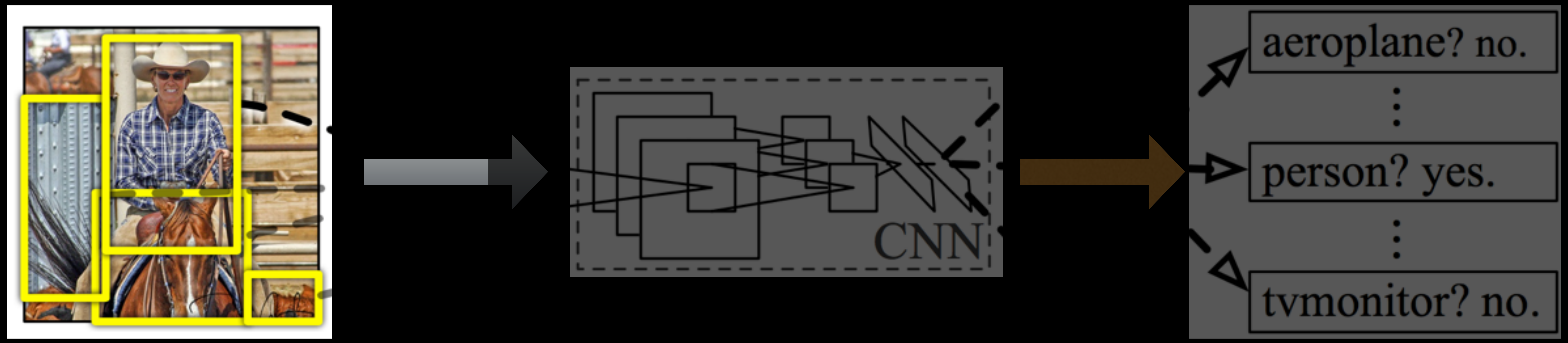
Figure 2: Recall *vs.* IoU overlap ratio on the PASCAL VOC 2007 test set.

Faster R-CNN Results

- Fewer and better proposals not only bring speed-up, but also detection performance boost.

method	# proposals	data	mAP (%)	time (ms)
SS	2k	07	66.9	1830
SS	2k	07+12	70.0	1830
RPN+VGG, unshared	300	07	68.5	342
RPN+VGG, shared	300	07	69.9	**196**
RPN+VGG, shared	300	07+12	**73.2**	**196**

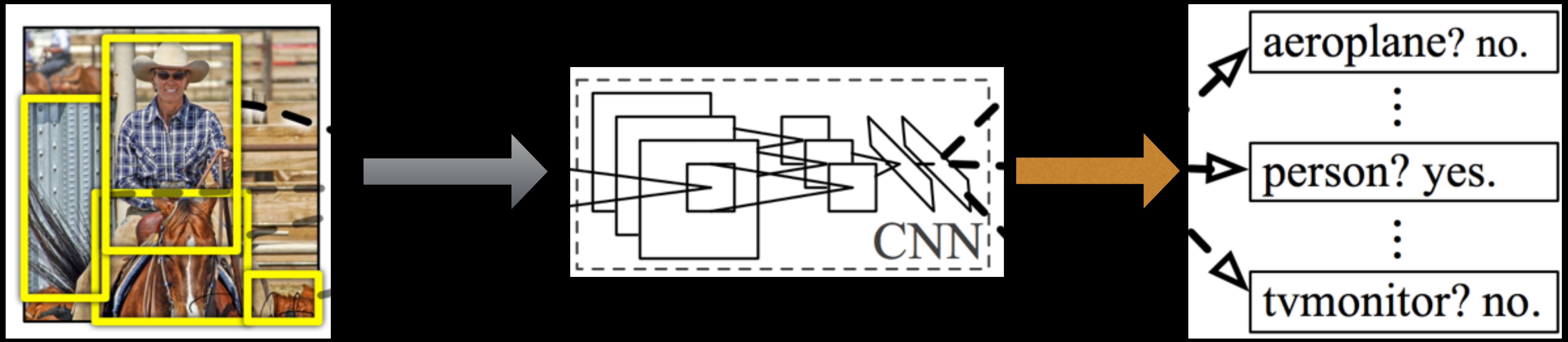
Object Detection System



Getting Proposals Feature Extraction Classifier

Faster R-CNN

Efficient Object Detection System



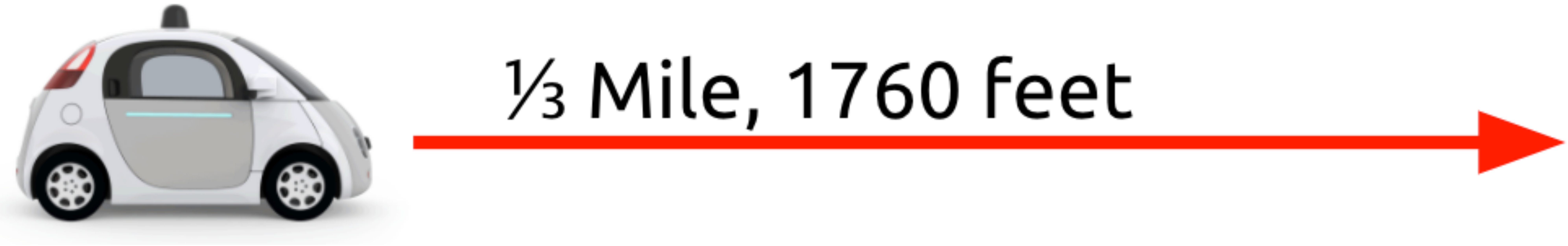
Getting Proposals	Feature Extraction	Classifier
Faster R-CNN	SPP	Fast R-CNN

66.0% —> 73.2%

47 s/im —> 0.2 s/im

Example 4: Driving car

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img



Slide credit: Joseph Chet Redmon

Example 4: Driving car

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
Fast R-CNN	70.0	.5 FPS	2 s/img

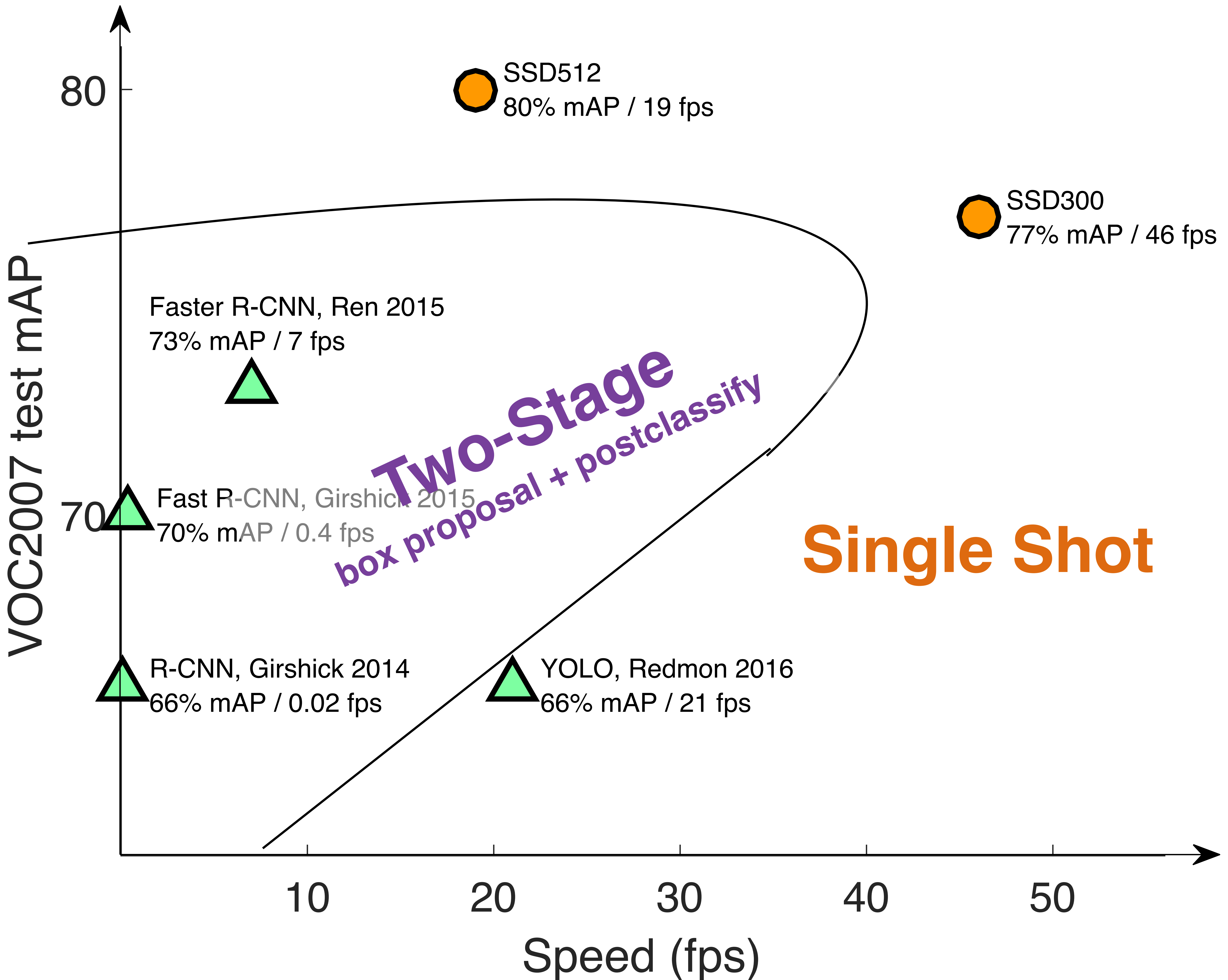
Slide credit: Joseph Chet Redmon

Example 4: Driving car

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
Fast R-CNN	70.0	.5 FPS	2 s/img
Faster R-CNN	73.2	7 FPS	140 ms/img

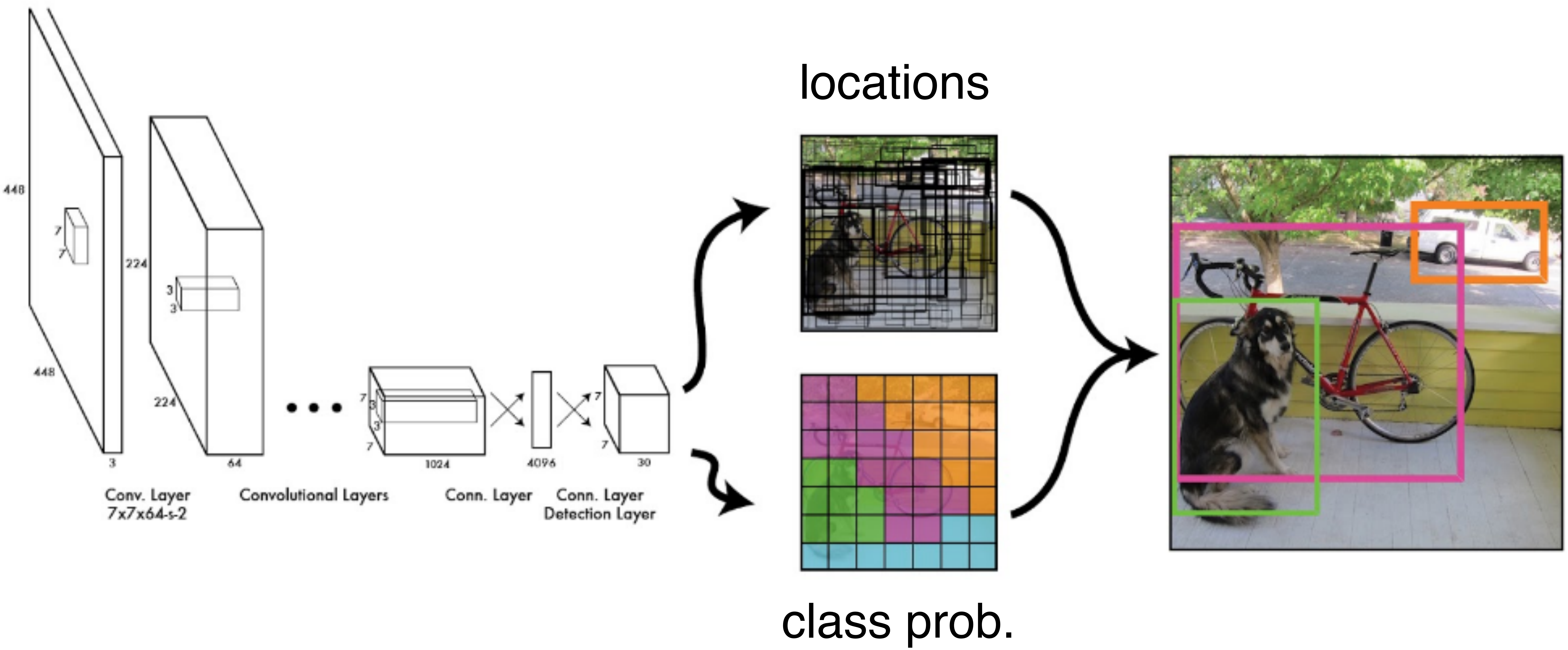
Slide credit: Joseph Chet Redmon

Real-time object detectors?



Slide credit: Wei Liu

YOLO: You Only Look Once

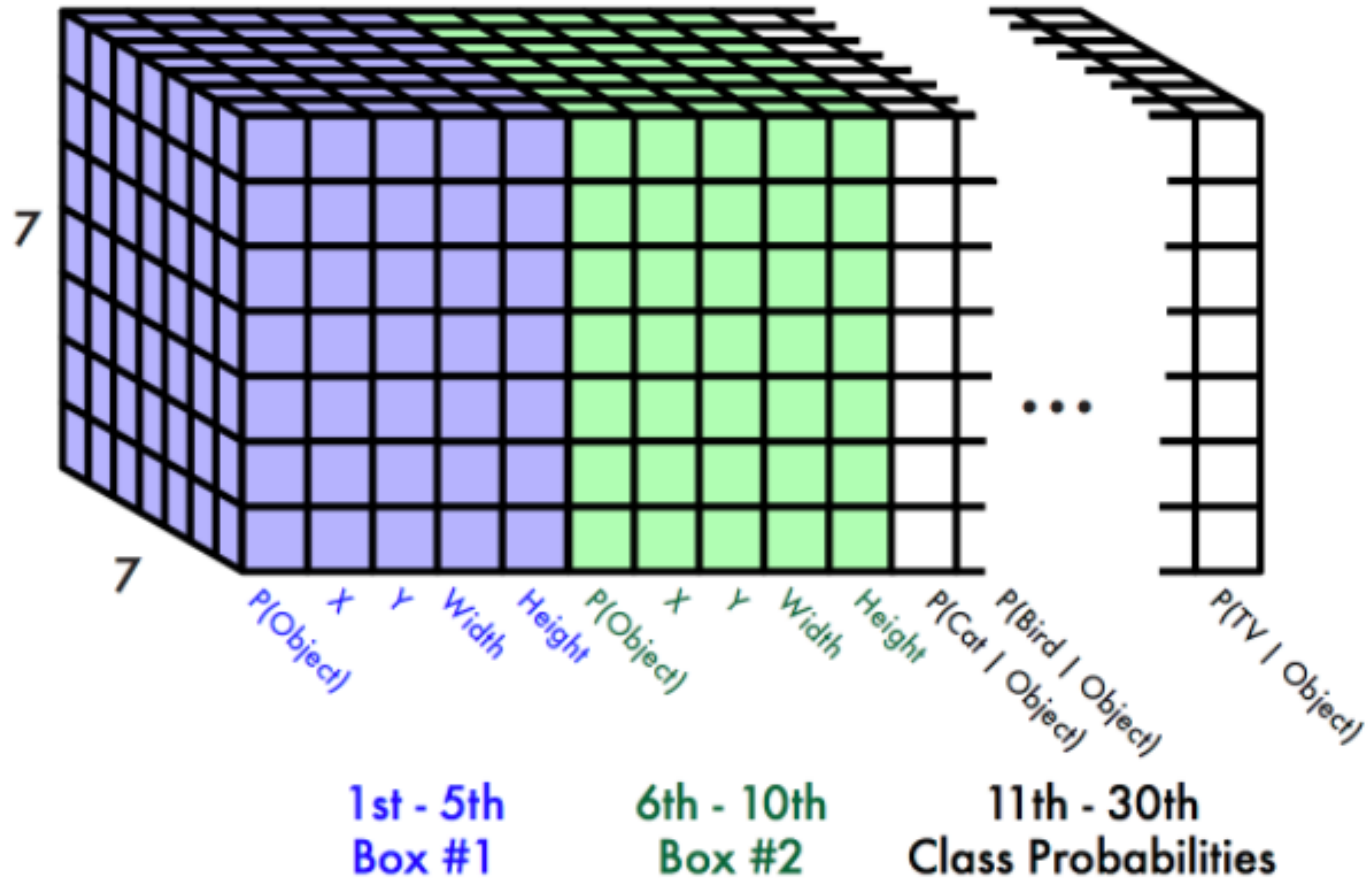


Slide credit: Redmon J, et al. You only look once: Unified, real-time object detection. CVPR2016

YOLO: output parameterization

Each cell predicts:

- For each bounding box:
  - 4 coordinates (x, y, w, h)
  - 1 confidence value
- Some number of class probabilities



For Pascal VOC:

- 7x7 grid
- 2 bounding boxes / cell
- 20 classes

7 x 7 x (2 x 5 + 20) = 7 x 7 x 30 tensor = **1470 outputs**

Slide credit: Redmon J, et al. You only look once: Unified, real-time object detection. CVPR2016

YOLO: limitations

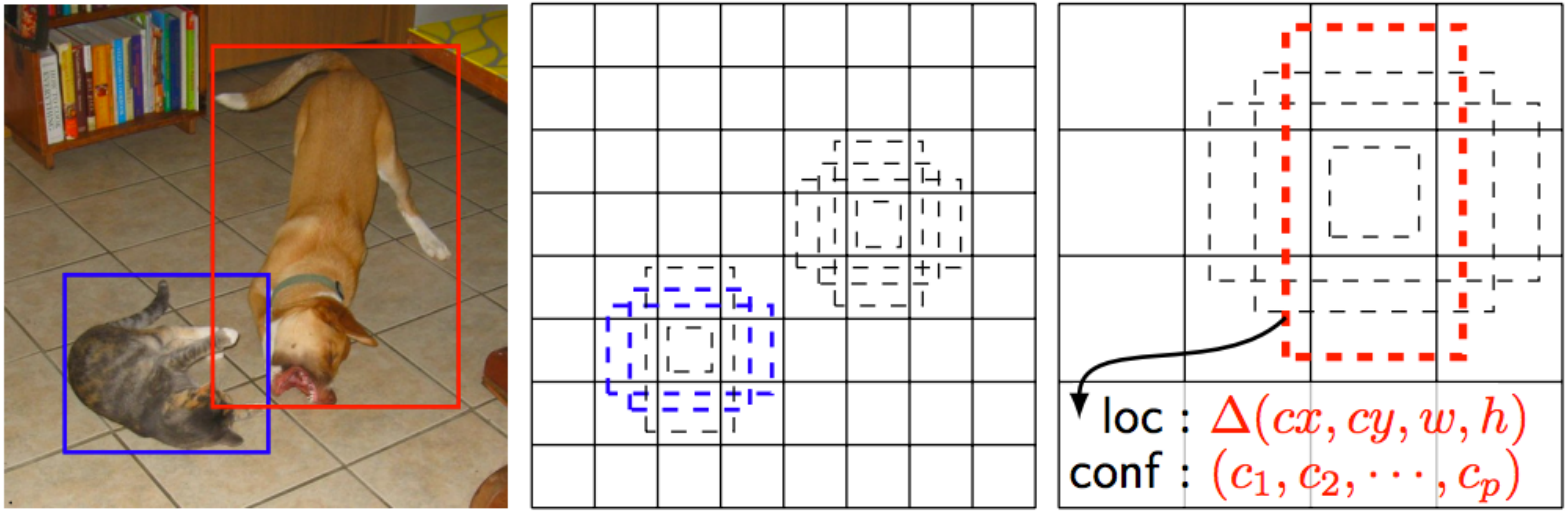
- Small objects
- Objects with different shapes/sizes
- Occluded objects

Slide credit: Redmon J, et al. You only look once: Unified, real-time object detection. CVPR2016

SSD: Single Shot MultiBox Detector

Slide credit: Wei L, et al. SSD: Single Shot MultiBox Detector. ECCV2016

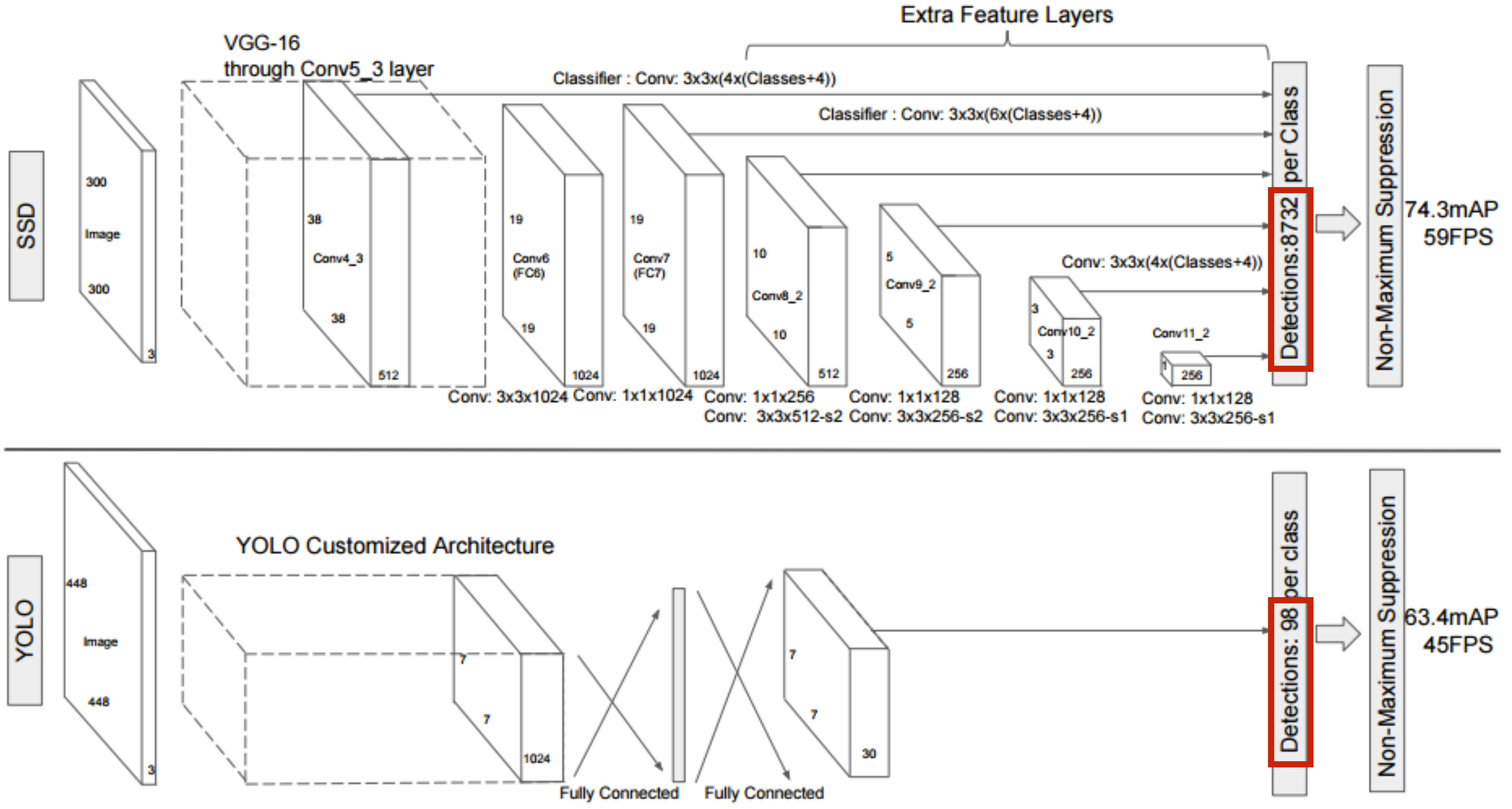
SSD: YOLO + default box shape + multi-scale



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Slide credit: Wei L, et al. SSD: Single Shot MultiBox Detector. ECCV2016

SSD: YOLO + default box shape + multi-scale



Slide credit: Wei L, et al. SSD: Single Shot MultiBox Detector. ECCV2016

Object detection

- Introduction
- Pre-CNN time
  - HOG detector
  - Deformable Part-based Model
- CNN time
  - Region CNN
  - Fast versions of RCNN
  - YOLO/SSD
- 3D object detection
- Devil's in the details

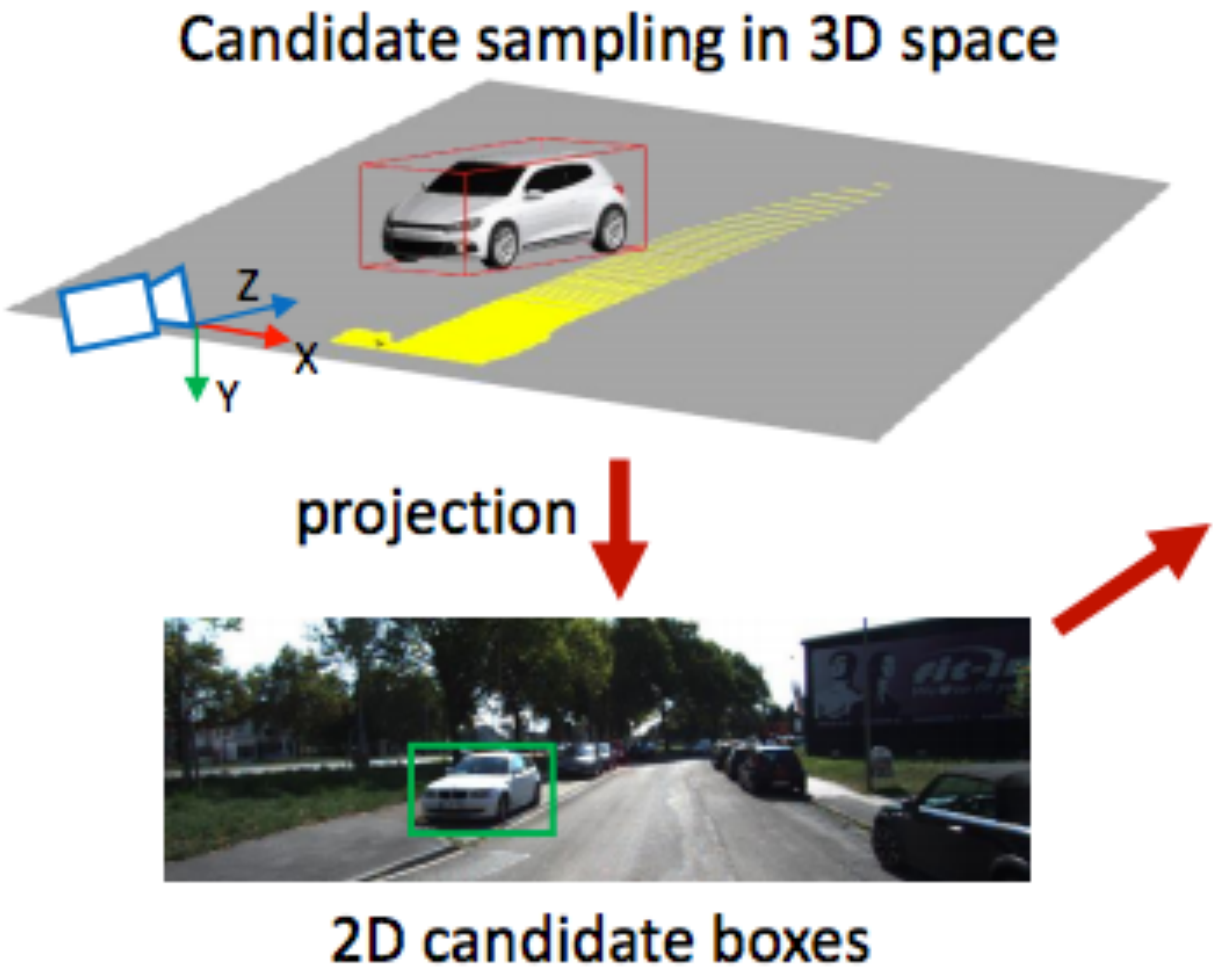
3D object detection: camera model

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\underbrace{w}_{\text{Scale factor}}\,[\underbrace{x\ y}_{\text{Image points}}\ 1] = [\underbrace{X\ Y\ Z}_{\text{World points}}\ 1]\ P$$

$$\underbrace{P}_{\text{Camera matrix}} = \underbrace{\begin{bmatrix} R \\ t \end{bmatrix}}_{\substack{\text{Extrinsics} \\ \text{Rotation and translation}}} \underbrace{K}_{\text{Intrinsic matrix}}$$

3D object detection: pipeline



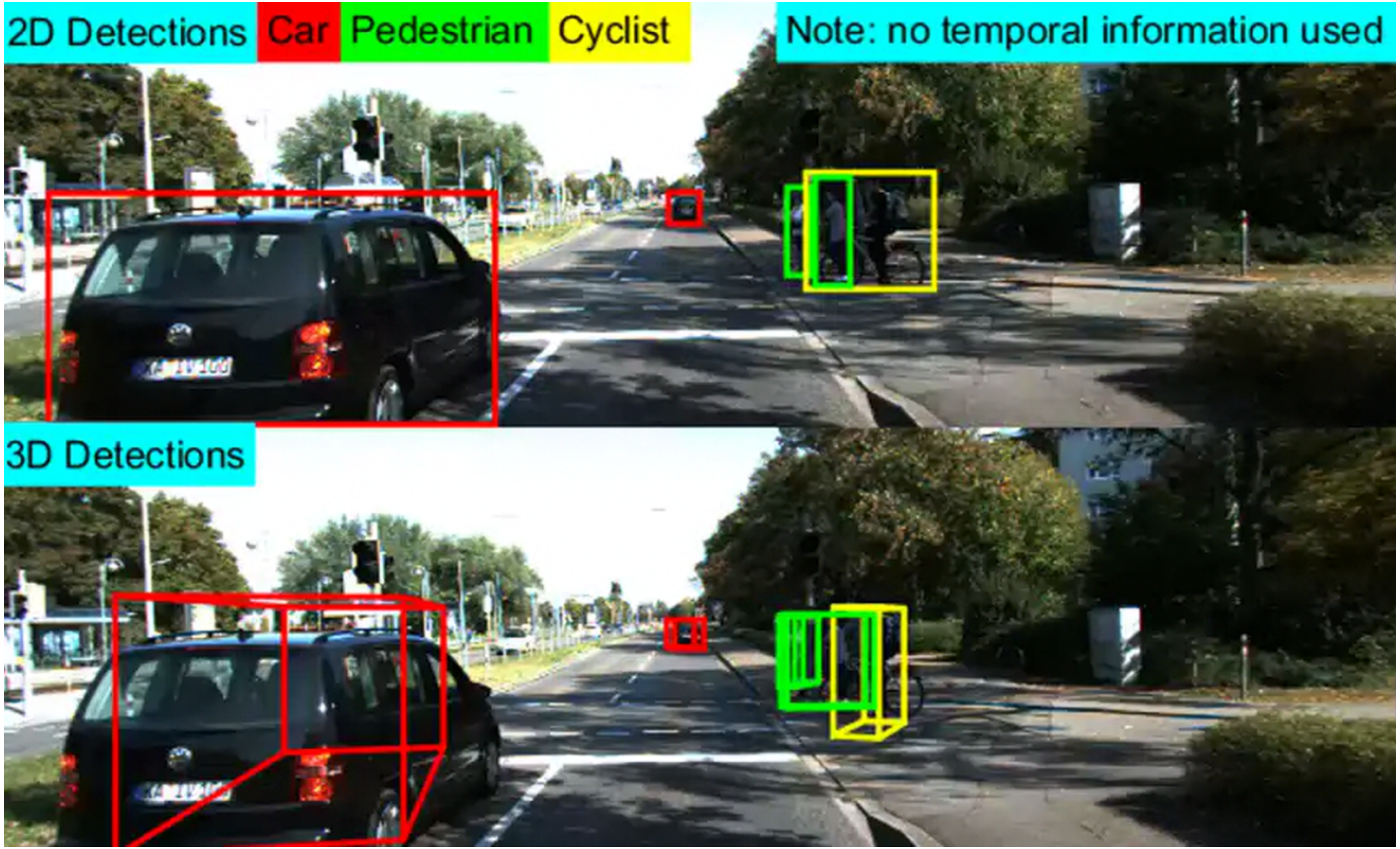
2D detector

3D NMS

3D detections

Chen X, et al. Monocular 3D Object Detection for Autonomous Driving. CVPR2016

3D object detection: demo



Chen X, et al. Monocular 3D Object Detection for Autonomous Driving. CVPR2016

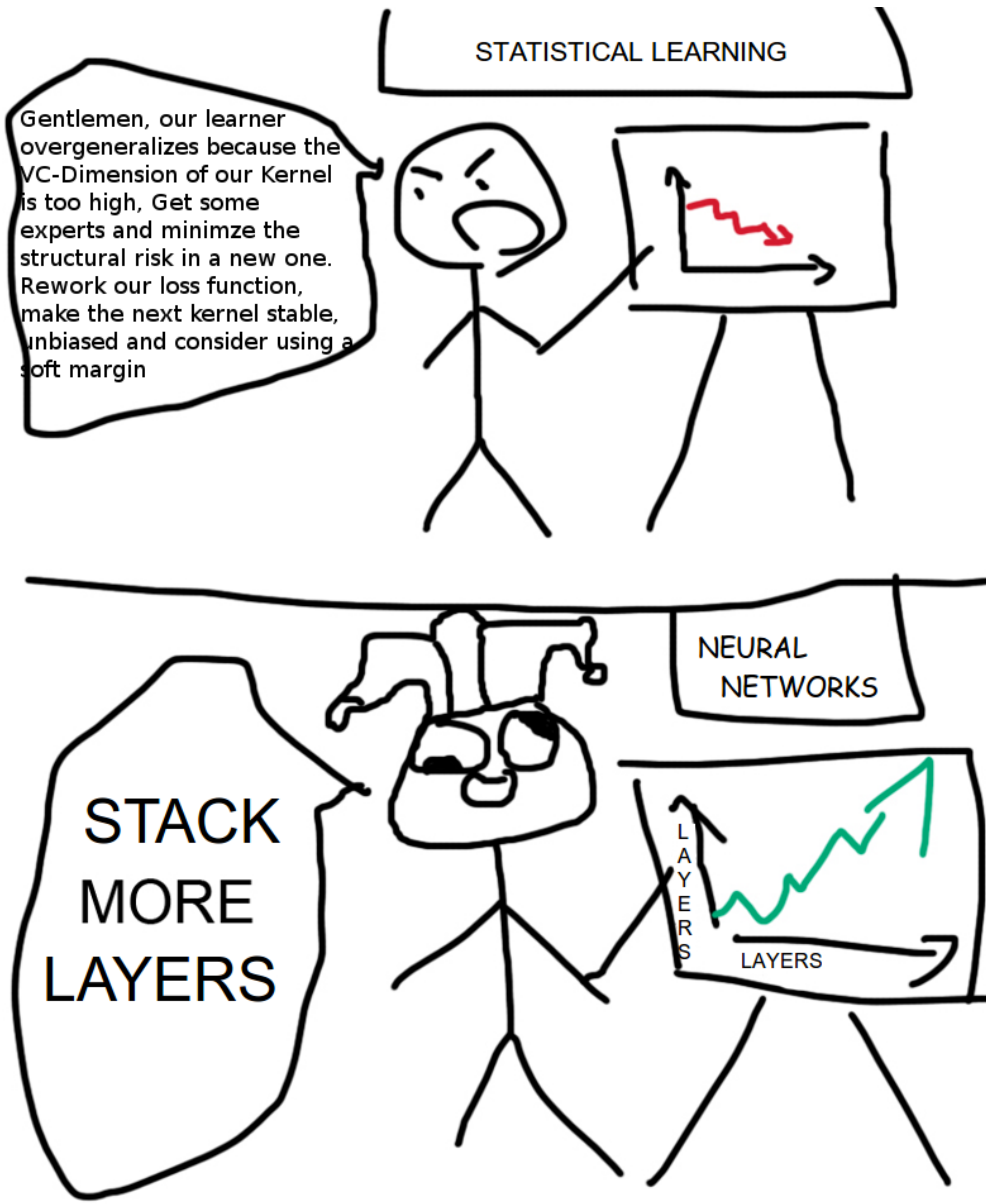
Object detection

- Introduction
- Pre-CNN time
  - HOG detector
  - Deformable Part-based Model
- CNN time
  - Region CNN
  - Fast versions of RCNN
  - YOLO/SSD
- 3D object detection
- Devil’s in the details

Trick: Pre-trained model

Faster R-CNN baseline	mAP@.5	mAP@.5:.95
VGG-16	41.5	21.5
ResNet-101	**48.4**	**27.2**

COCO detection results
(ResNet has 28% relative gain)



He K, et al. Deep Residual Learning for Image Recognition. CVPR2016

Trick: Sampling

1. Use 'ignore' labels:

Difficulties are defined as follows:

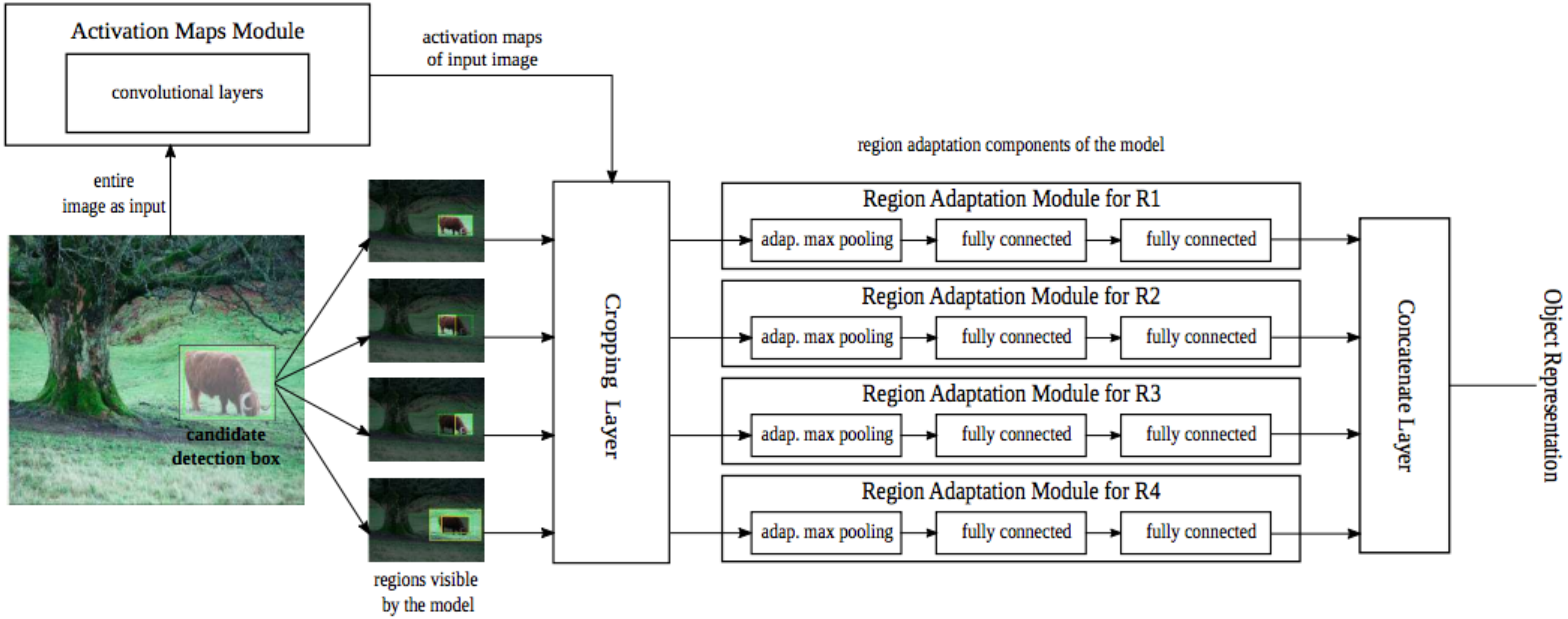
- **Easy:** Min. bounding box height: 40 Px, Max. occlusion level: Fully visible, Max. truncation: 15 %
- **Moderate:** Min. bounding box height: 25 Px, Max. occlusion level: Partly occluded, Max. truncation: 30 %
- **Hard:** Min. bounding box height: 25 Px, Max. occlusion level: Difficult to see, Max. truncation: 50 %

2. Use hard-example mining:

- Heuristics
- Offline
- Online[1]

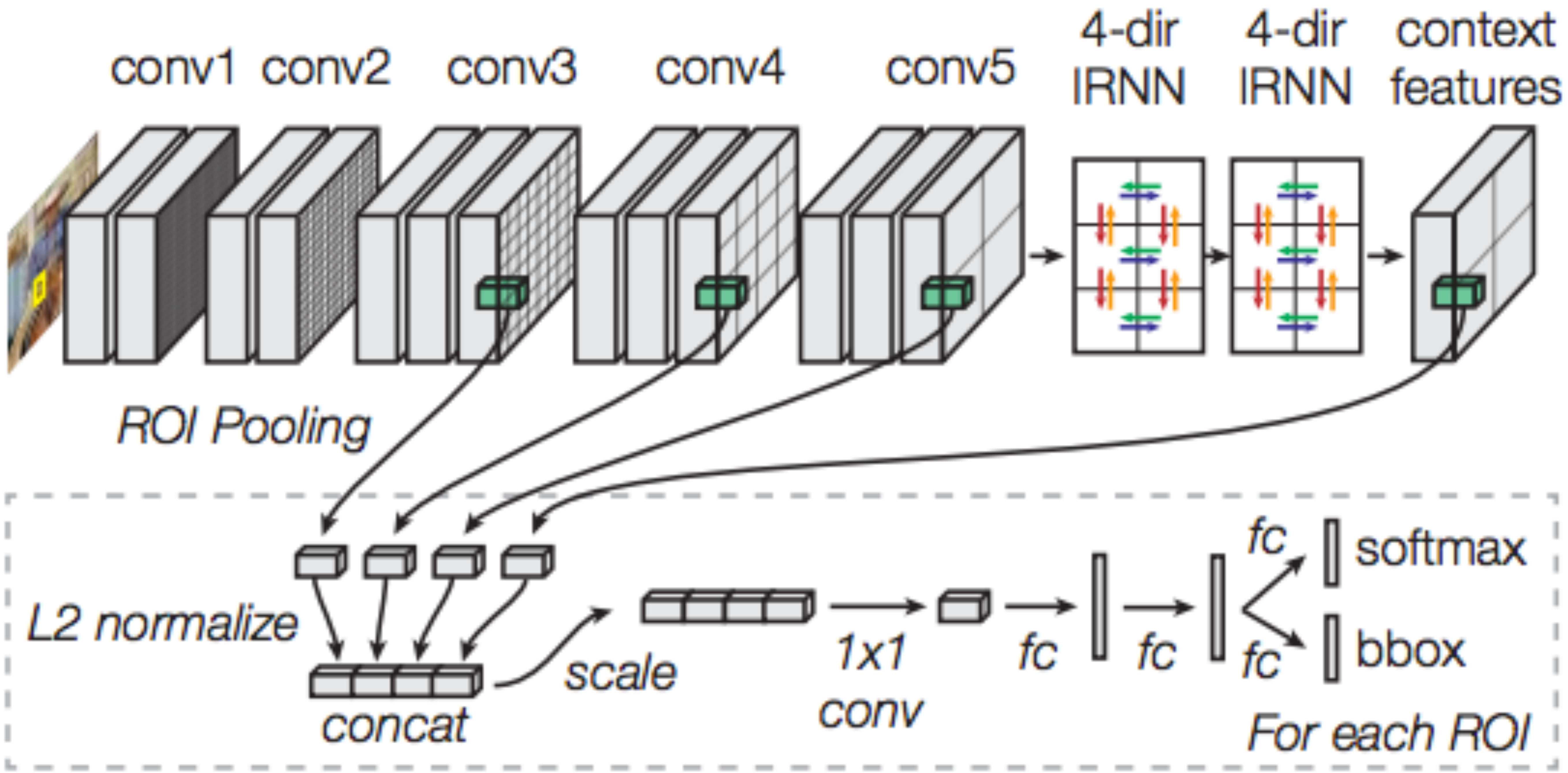
[1] Shrivastava A, et al. Training region-based object detectors with online hard example mining. CVPR2016

Trick: Multi-region ensemble



Gidaris S, et al. Object detection via a multi-region and semantic segmentation-aware cnn model. ICCV2015

Trick: Multi-scale feature fusion



Bell S, et al. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. CVPR2016

Trick: Iterative localization

- Iterative bounding box regression
- Voting NMS

(a) Step 1 (b) Step 2 (c) Step 3 (d) Step 4 (e) Step 5

Gidaris S, et al. Object detection via a multi-region and semantic segmentation-aware cnn model. ICCV2015

Object detection

- Introduction
- Pre-CNN time
  - HOG detector
  - Deformable Part-based Model
- CNN time
  - Region-CNN
  - Fast versions of R-CNN
  - YOLO/SSD
- 3D object detection
- Devil's in the details

we need features!
we need flexible models!

we need better features!

we want to be fast!

we want to be real-time!

we like 3D!

we hack!

Q&A

“The only stupid question is the one you never asked.” - Rich Sutton