Intro to Object Detection

CSC2548, 2018 Winter
Bin Yang
17 Jan. 2018

"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

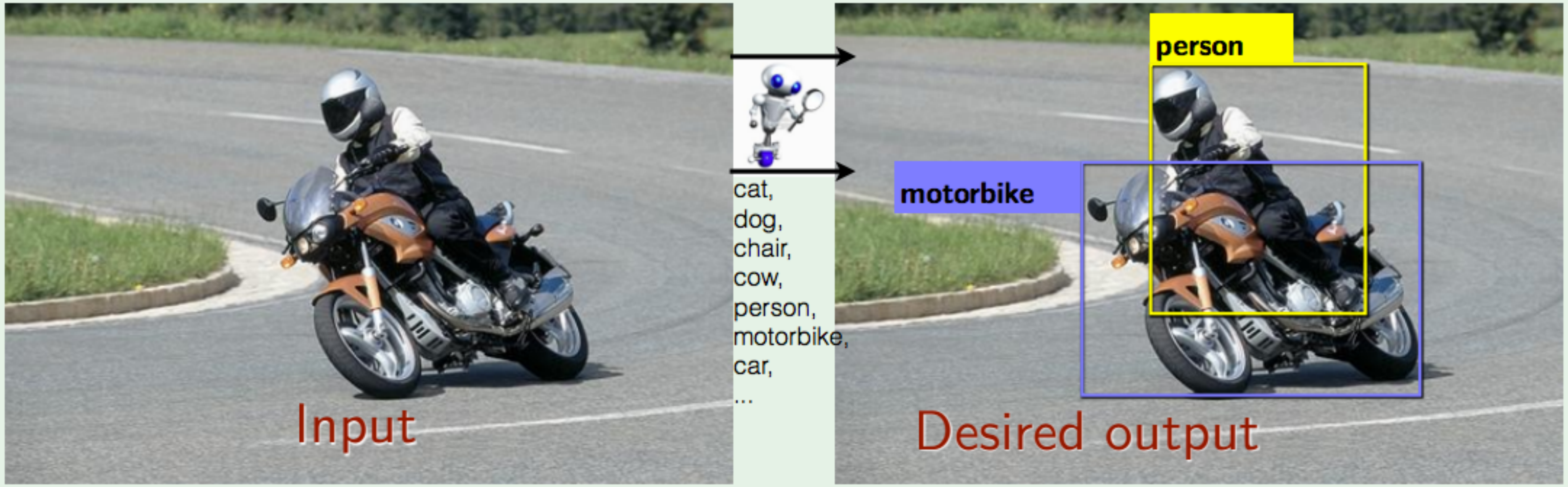
- Introduction
- Face detection: from Viola-Jones to CNN
- General object detection
    - HOG detector
    - Deformable Part-based Model
    - Region-CNN
    - Fast versions of R-CNN
    - YOLO/SSD
- Future directions

Object detection

- Introduction
- Face detection: from Viola-Jones to CNN
- General object detection
  - HOG detector
  - Deformable Part-based Model
  - Region-CNN
  - Fast versions of R-CNN
  - YOLO/SSD
- Future directions

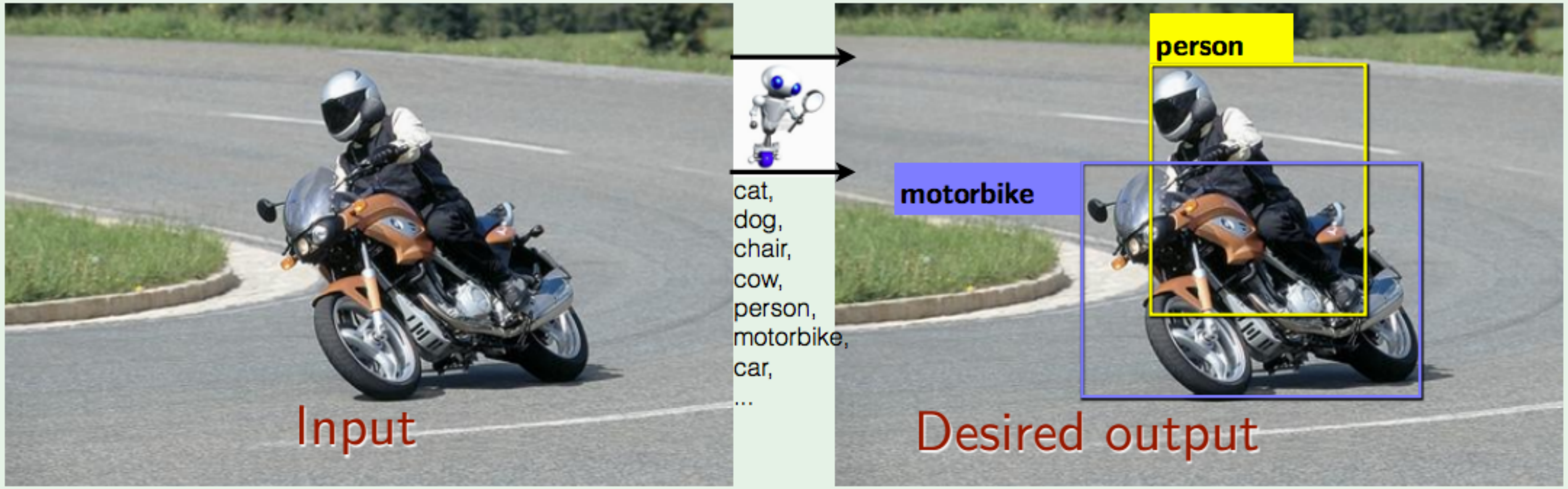
Formalizing the object detection task

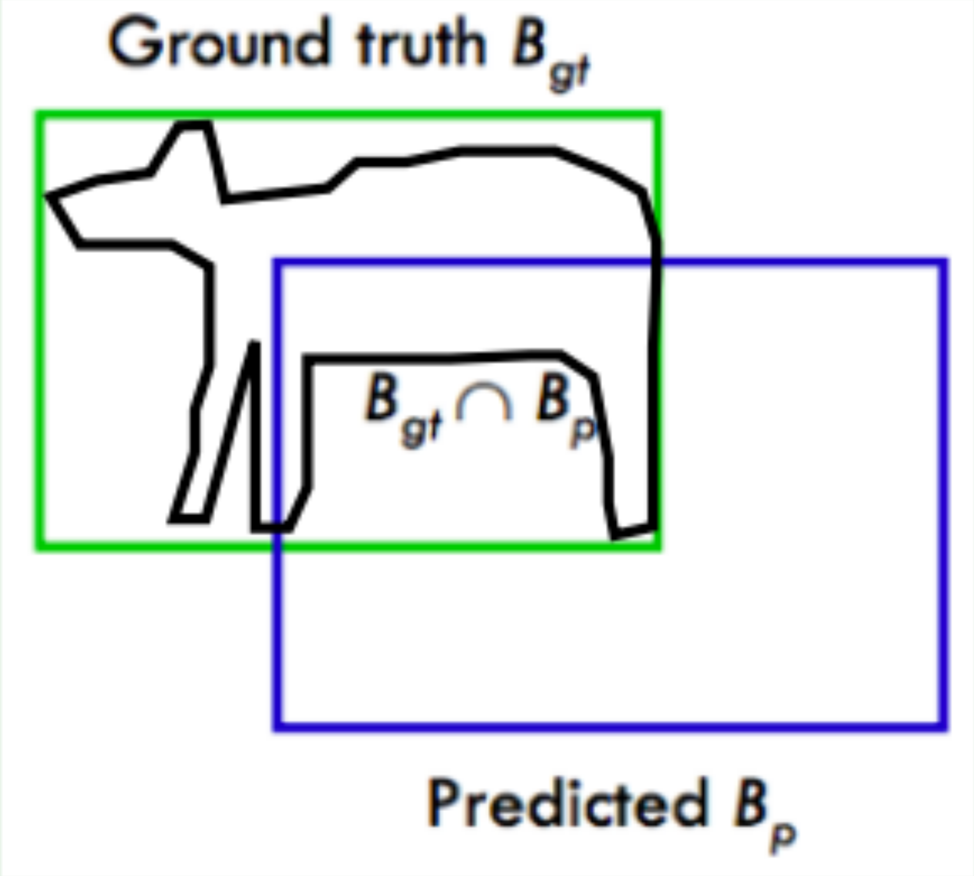
Many possible ways, this one is popular:

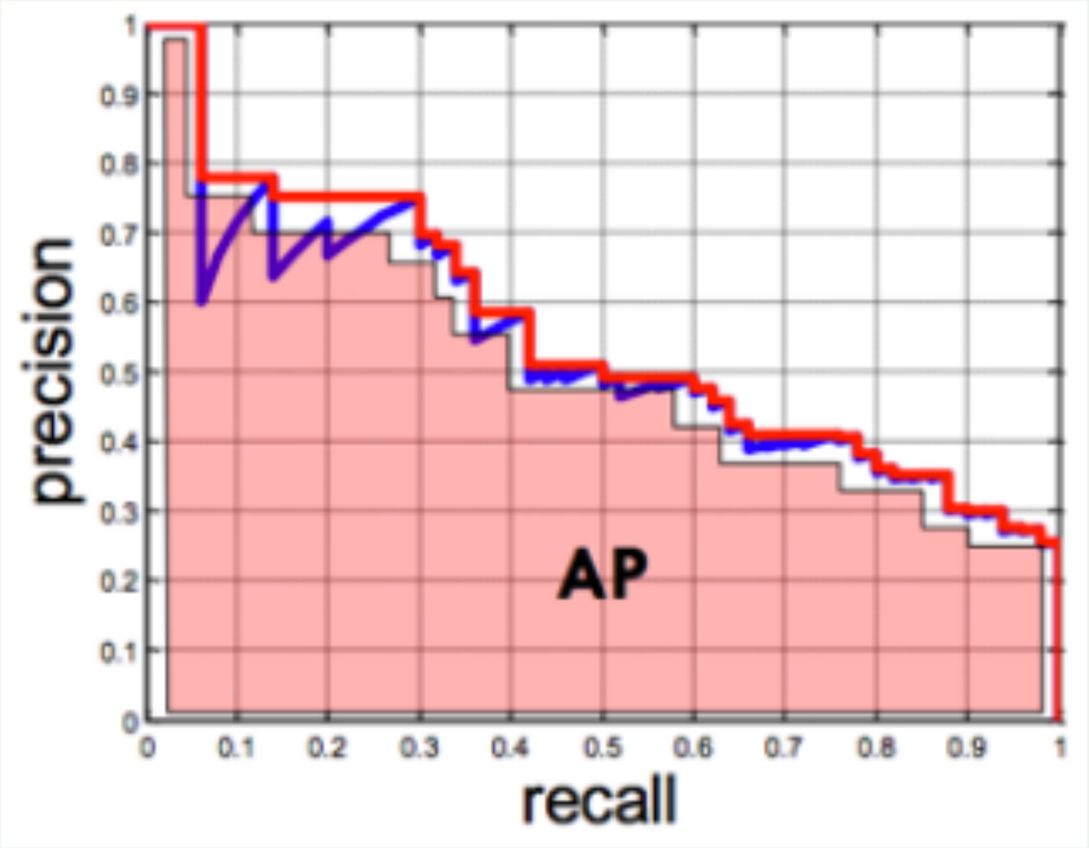


Formalizing the object detection task

Many possible ways, this one is popular:







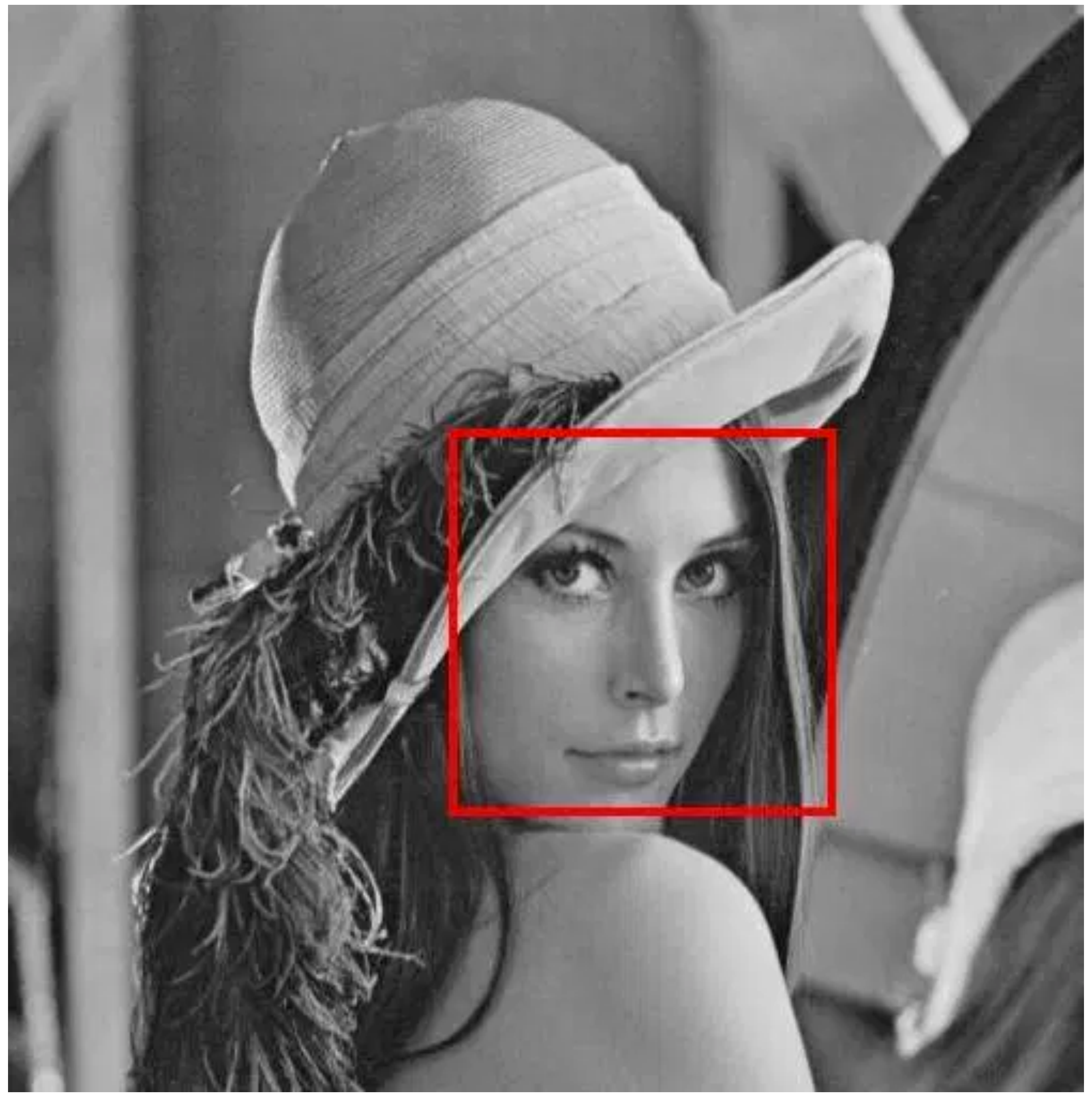
Performance summary:

Average Precision (AP)
0 is worst 1 is perfect

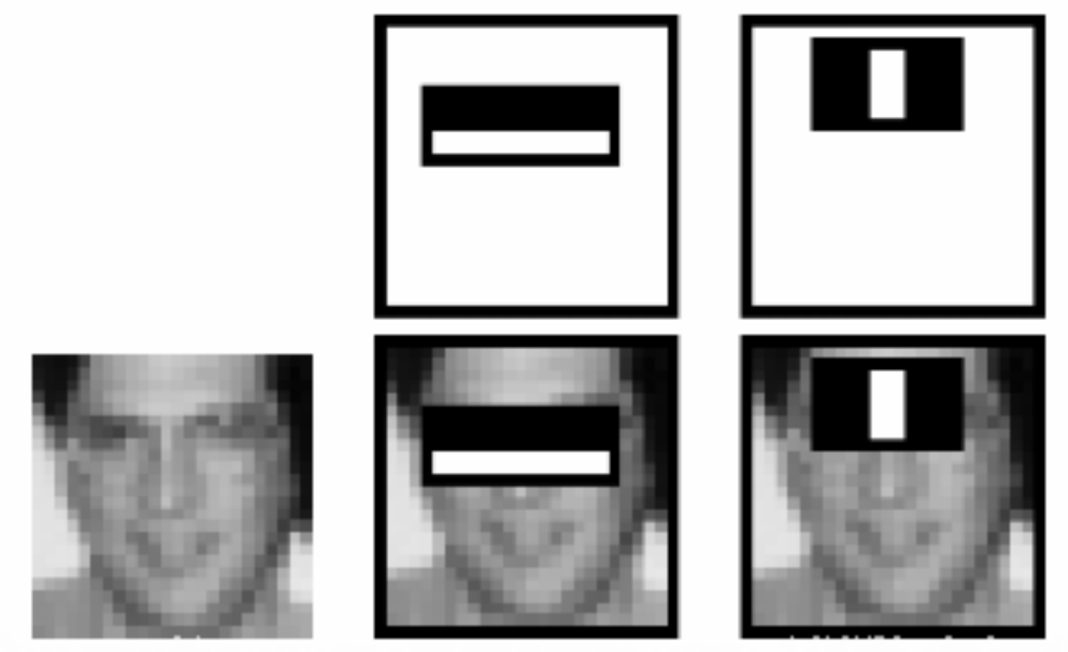
slide credit: Ross Girshick

Object detection

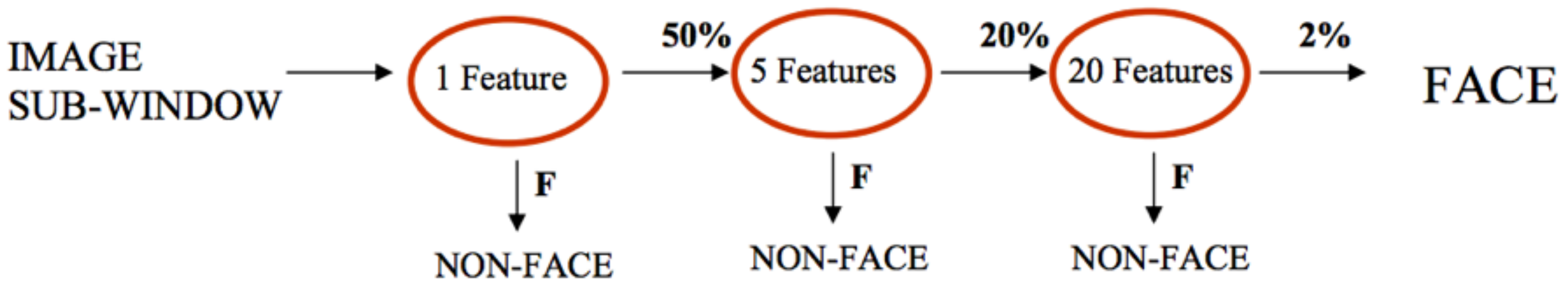
- Introduction
- Face detection: from Viola-Jones to CNN
- General object detection
  - HOG detector
  - Deformable Part-based Model
  - Region-CNN
  - Fast versions of R-CNN
  - YOLO/SSD
- Future directions



Viola/Jones face detector (2001, The Longuet-Higgins Prize in 2011)



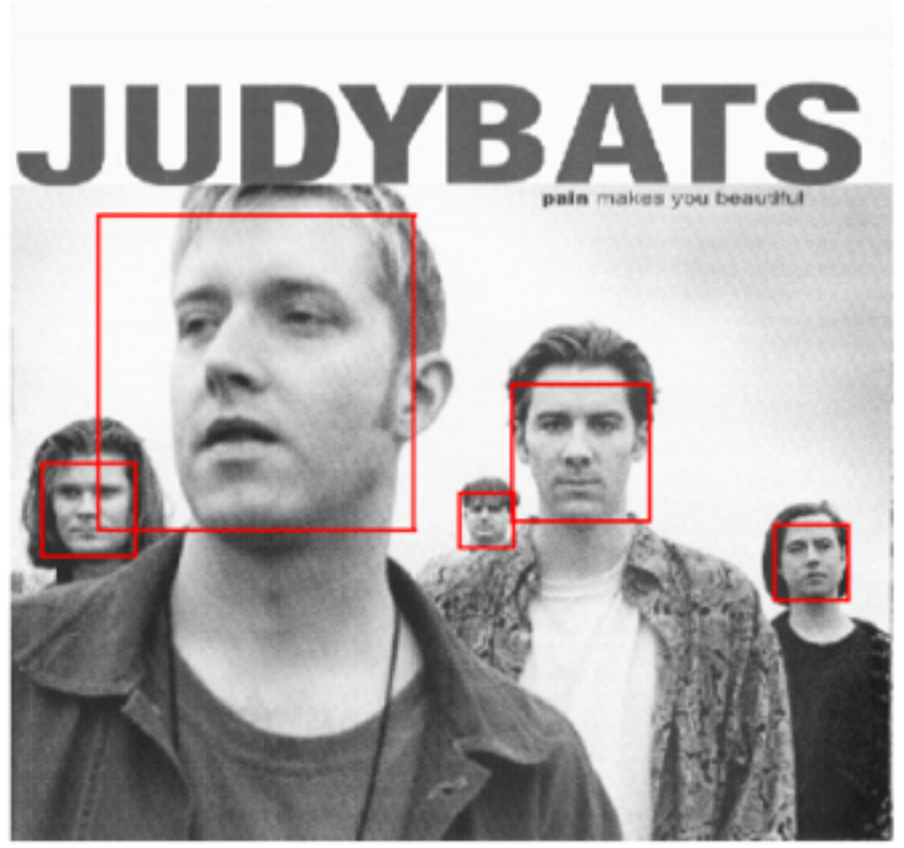
Cascaded Classifier

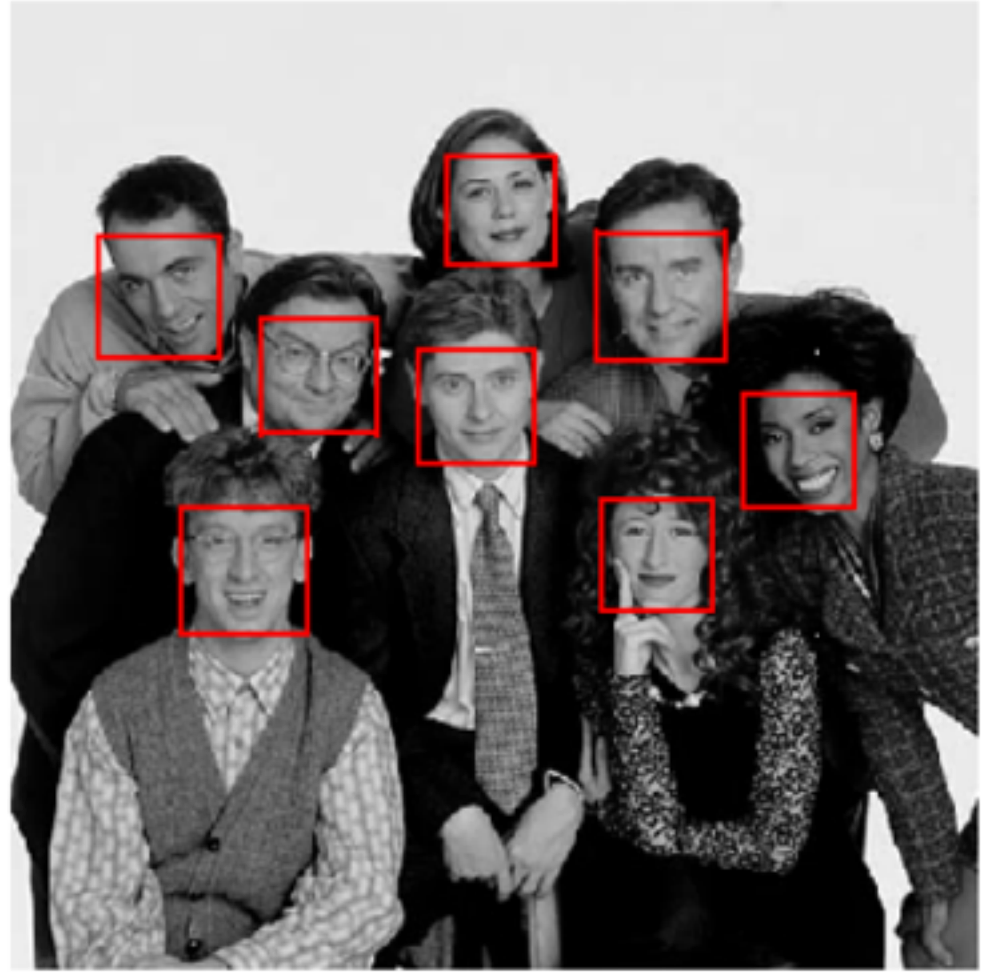


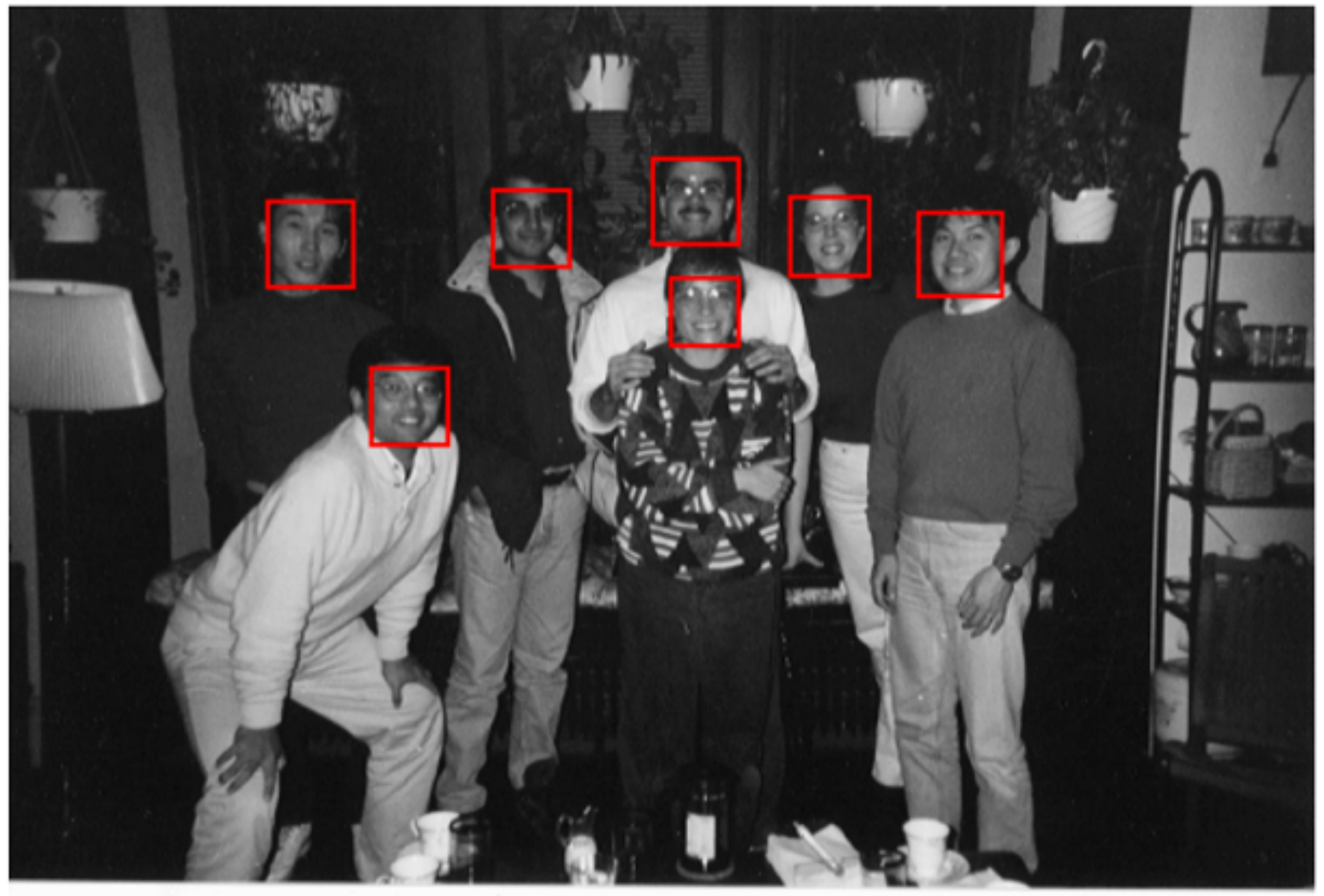
- A 1 feature classifier achieves 100% detection rate and about 50% false positive rate.
- A 5 feature classifier achieves 100% detection rate and 40% false positive rate (20% cumulative)
  - using data from previous stage.
- A 20 feature classifier achieve 100% detection rate with 10% false positive rate (2% cumulative)

slide from David Lowe

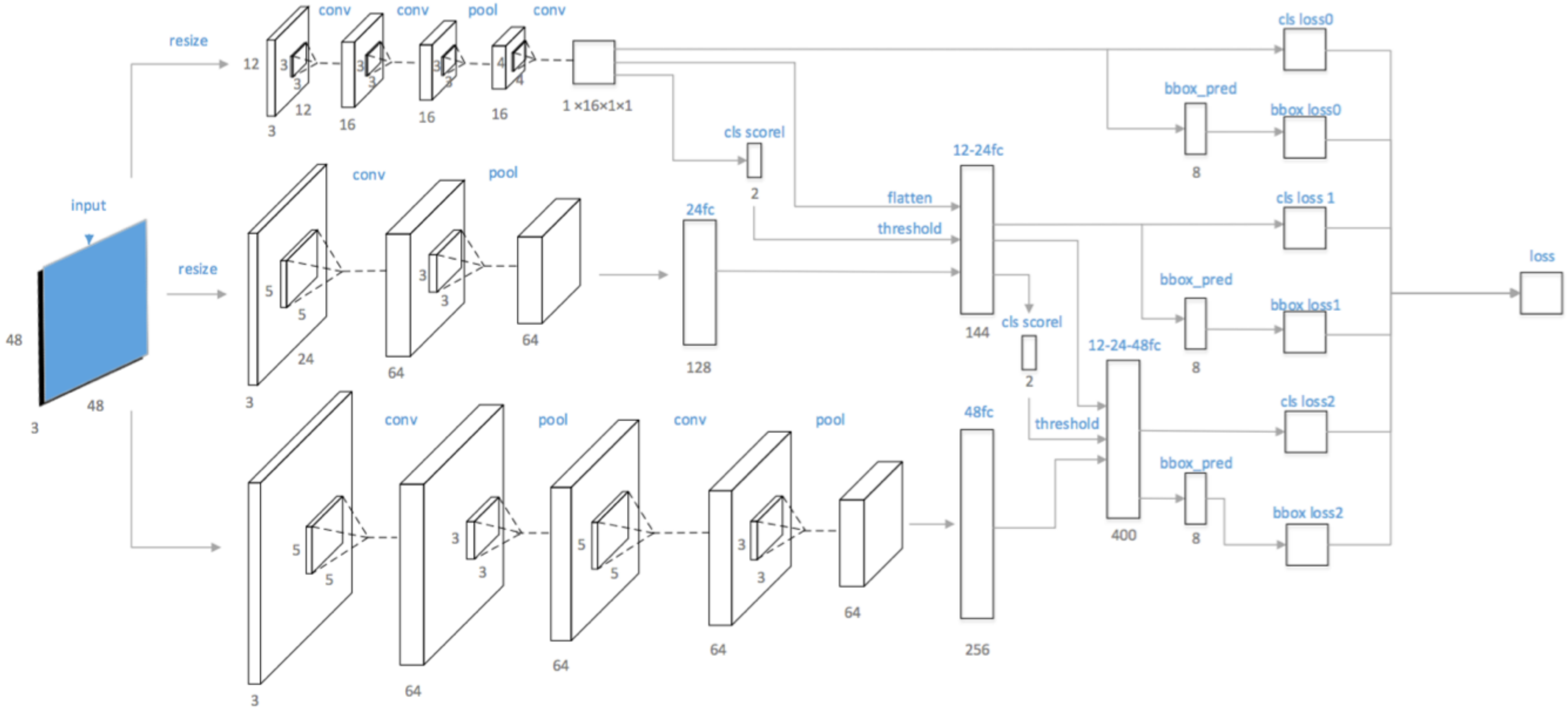
VJ face detection results







CNN based face detector (H. Qin, 2016)



Demo

Reference: http://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Qin_Joint_Training_of_CVPR_2016_paper.pdf

Object detection

- Introduction
- Face detection: from Viola-Jones to CNN
- General object detection
  - HOG detector
  - Deformable Part-based Model
  - Region-CNN
  - Fast versions of R-CNN
  - YOLO/SSD
- Future directions

The HOG Detector

N. Dalal and B. Triggs

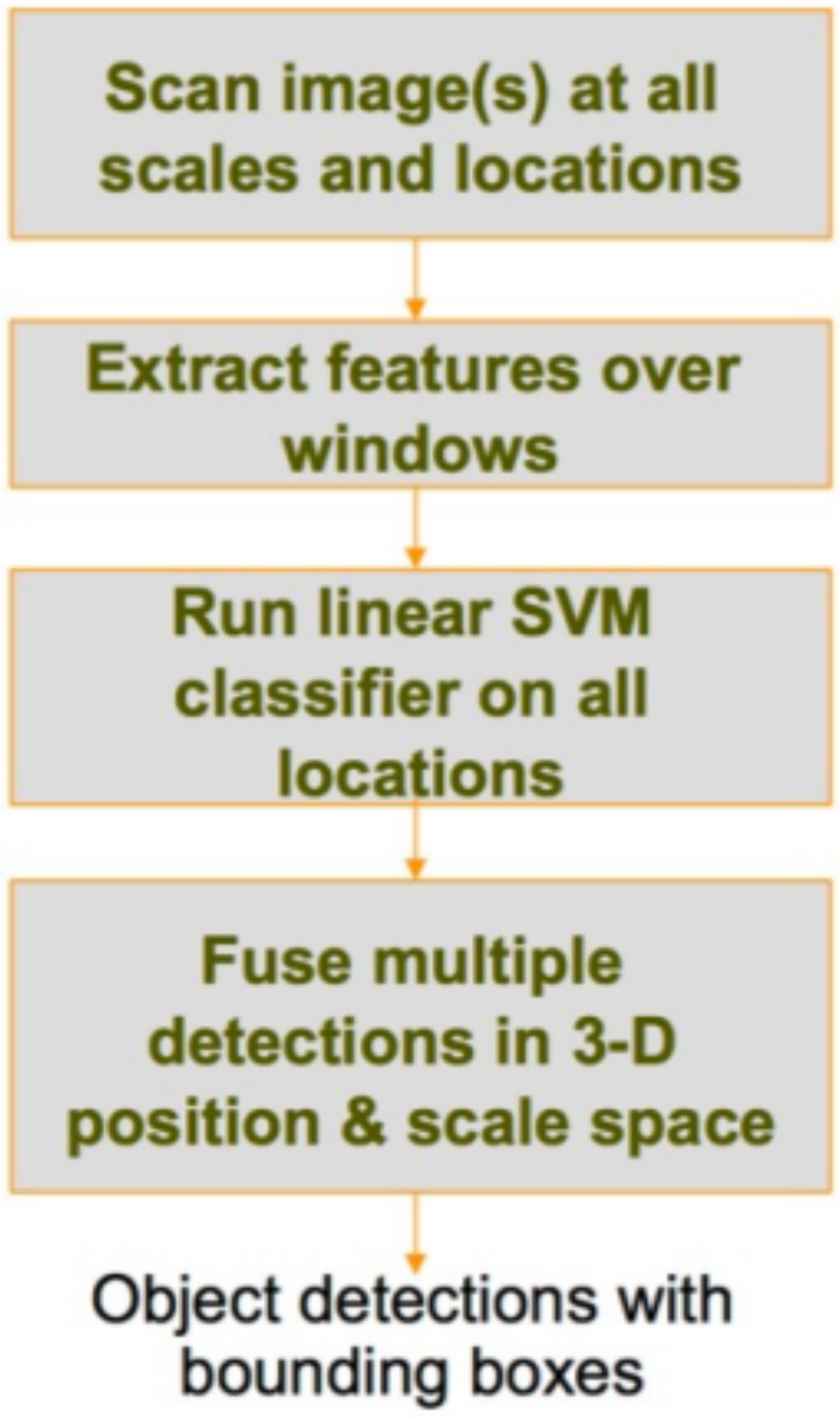
Histograms of oriented gradients for human detection

CVPR, 2005

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

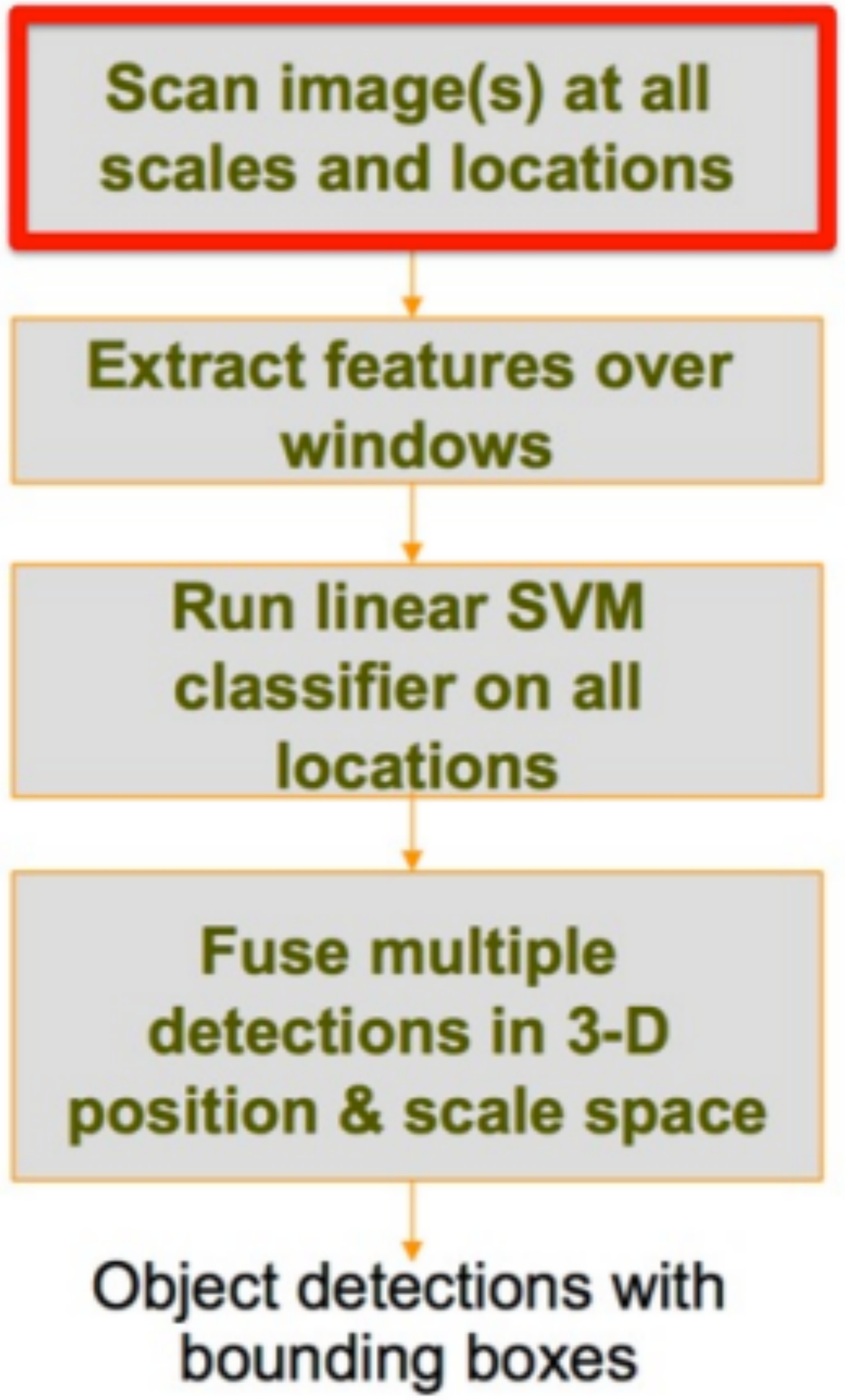
cited by 17,502

HOG detector: pipeline

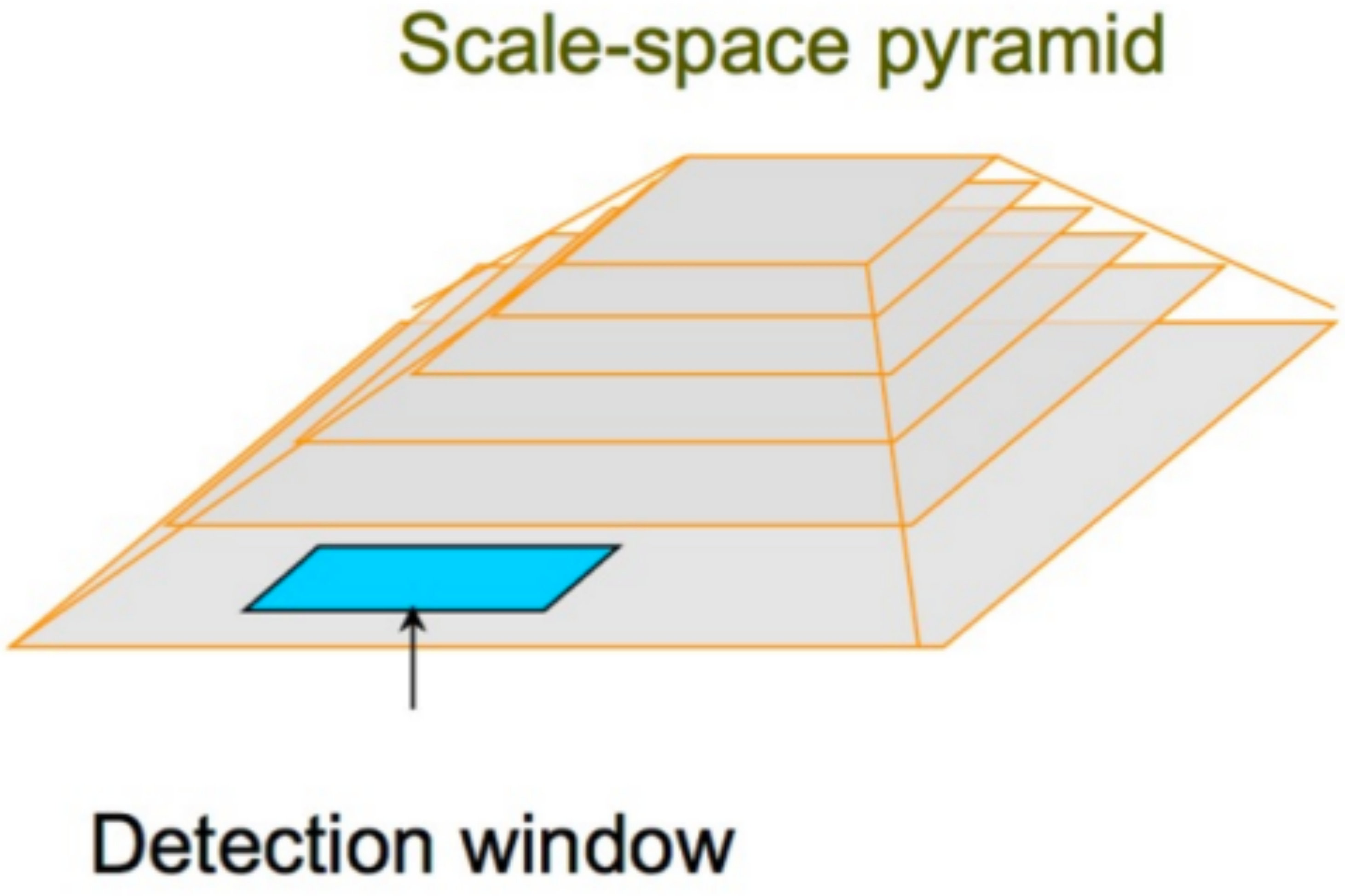


Slide credit: Sanja Fidler

I. Sliding window



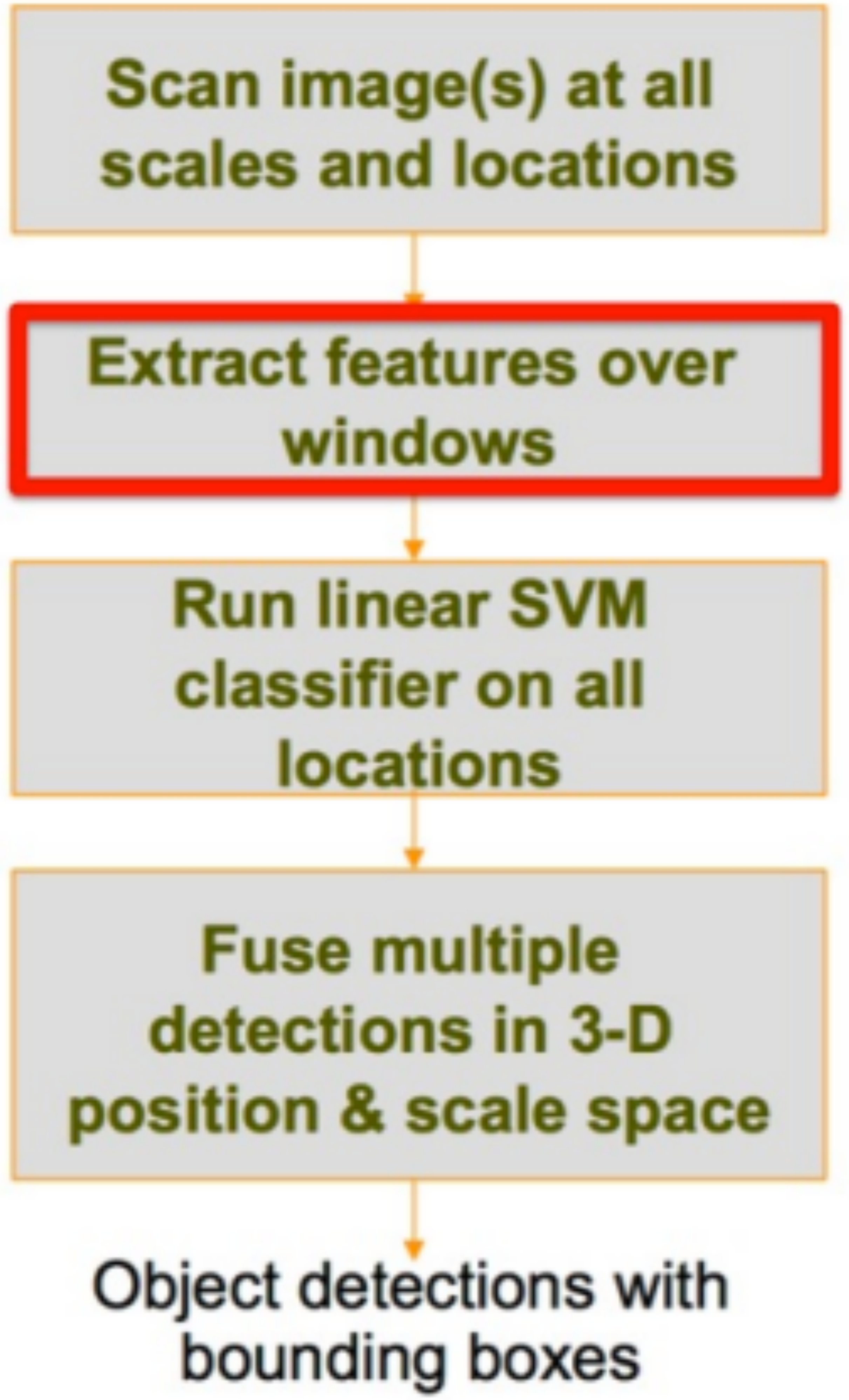
locations

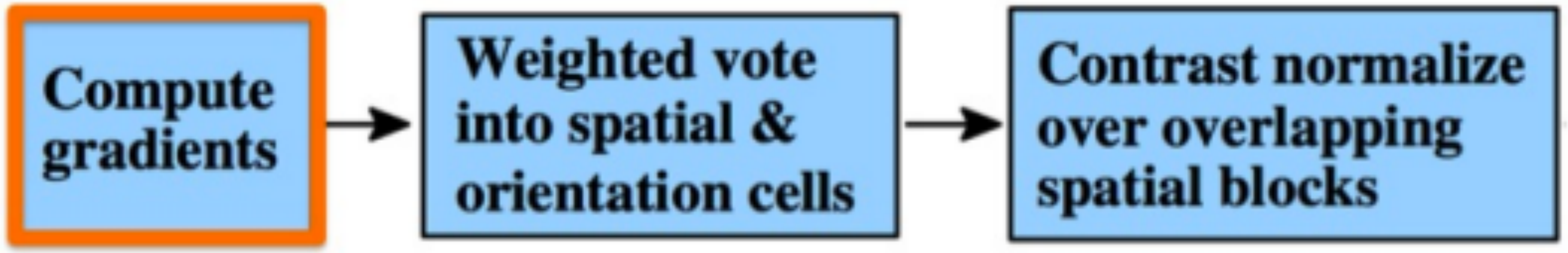


scales

Slide credit: Sanja Fidler

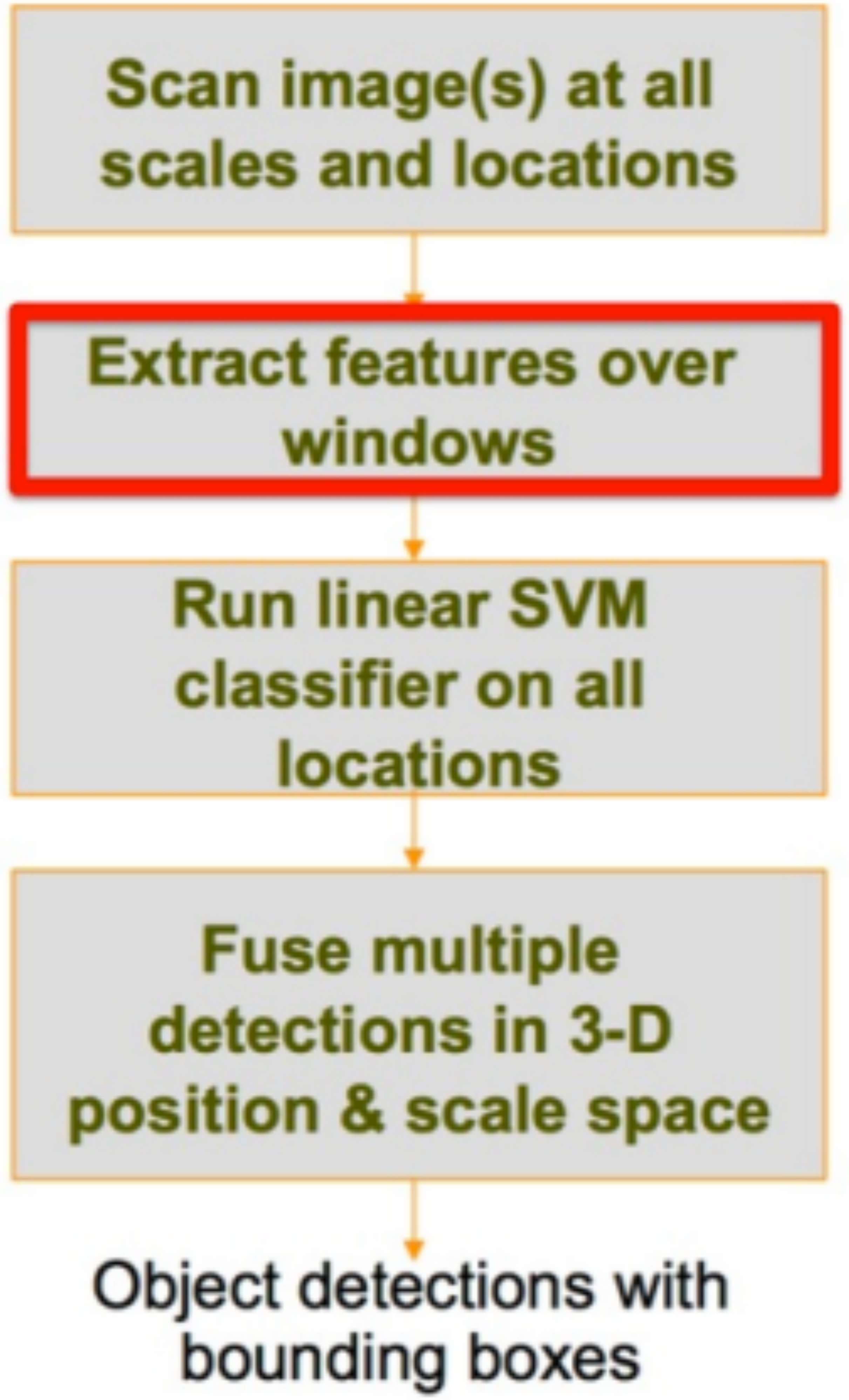
II. Histograms of Oriented Gradients

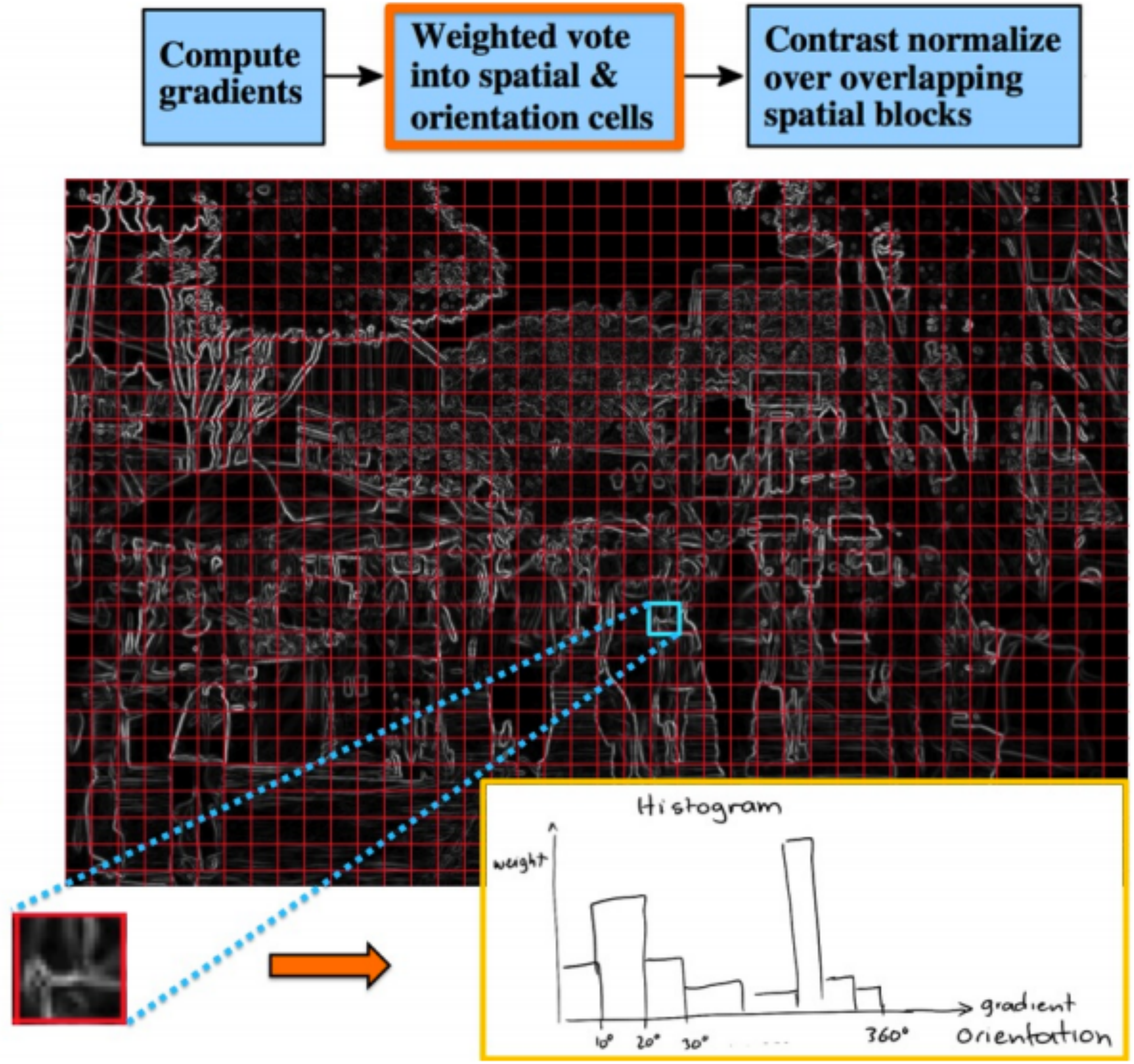




Slide credit: Sanja Fidler

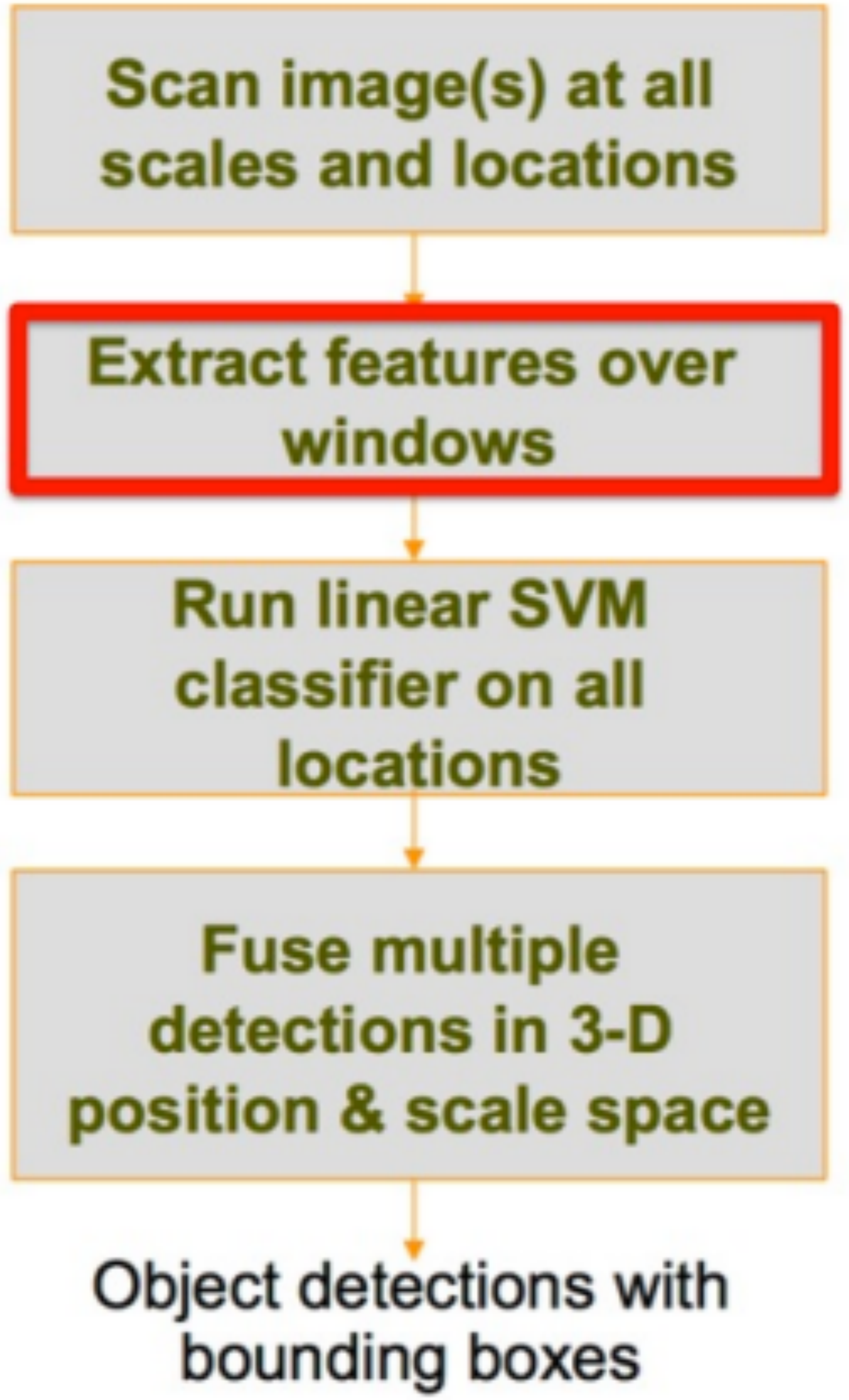
II. Histograms of Oriented Gradients

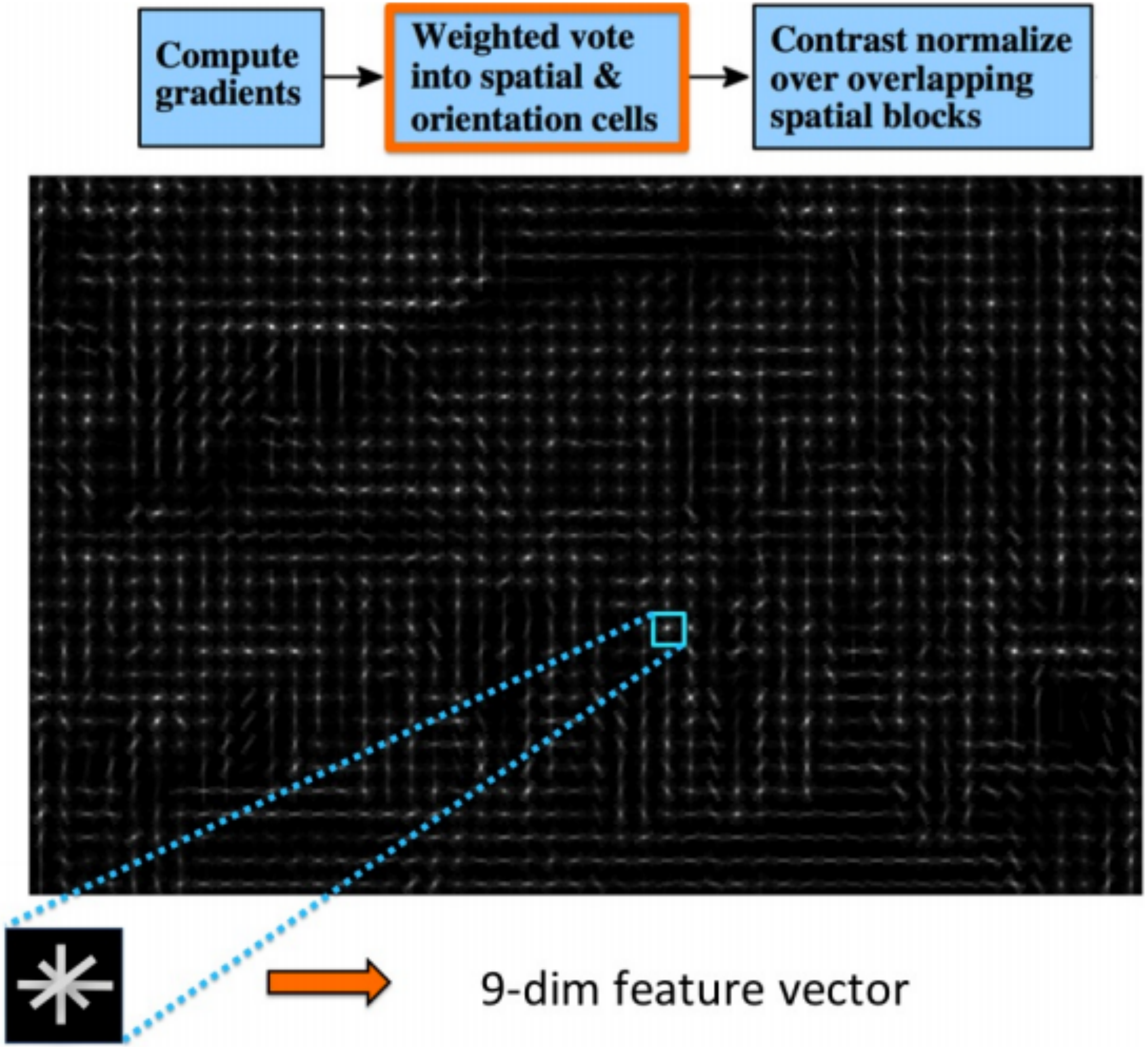




Slide credit: Sanja Fidler

II. Histograms of Oriented Gradients





Slide credit: Sanja Fidler

II. Histograms of Oriented Gradients

Scan image(s) at all scales and locations

↓

Extract features over windows

↓

Run linear SVM classifier on all locations

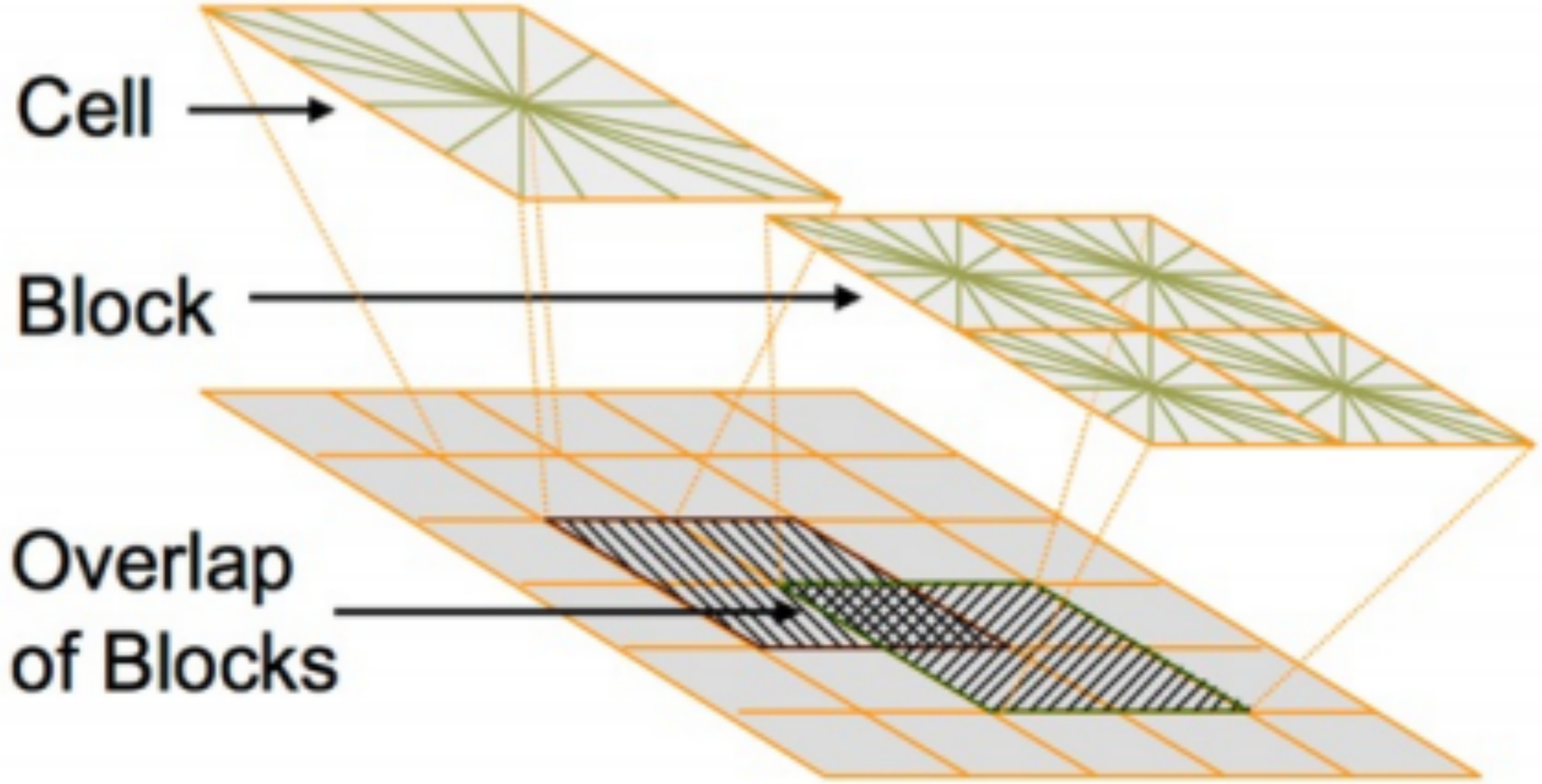
↓

Fuse multiple detections in 3-D position & scale space

↓

Object detections with bounding boxes

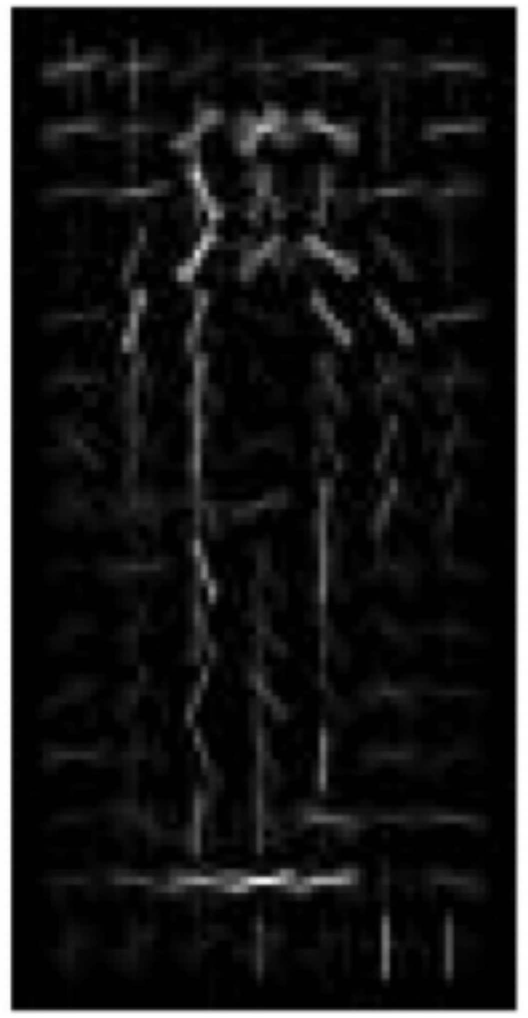
Compute gradients → Weighted vote into spatial & orientation cells → **Contrast normalize over overlapping spatial blocks**



Feature vector $f = [\ ..., \ ..., \quad ...]$

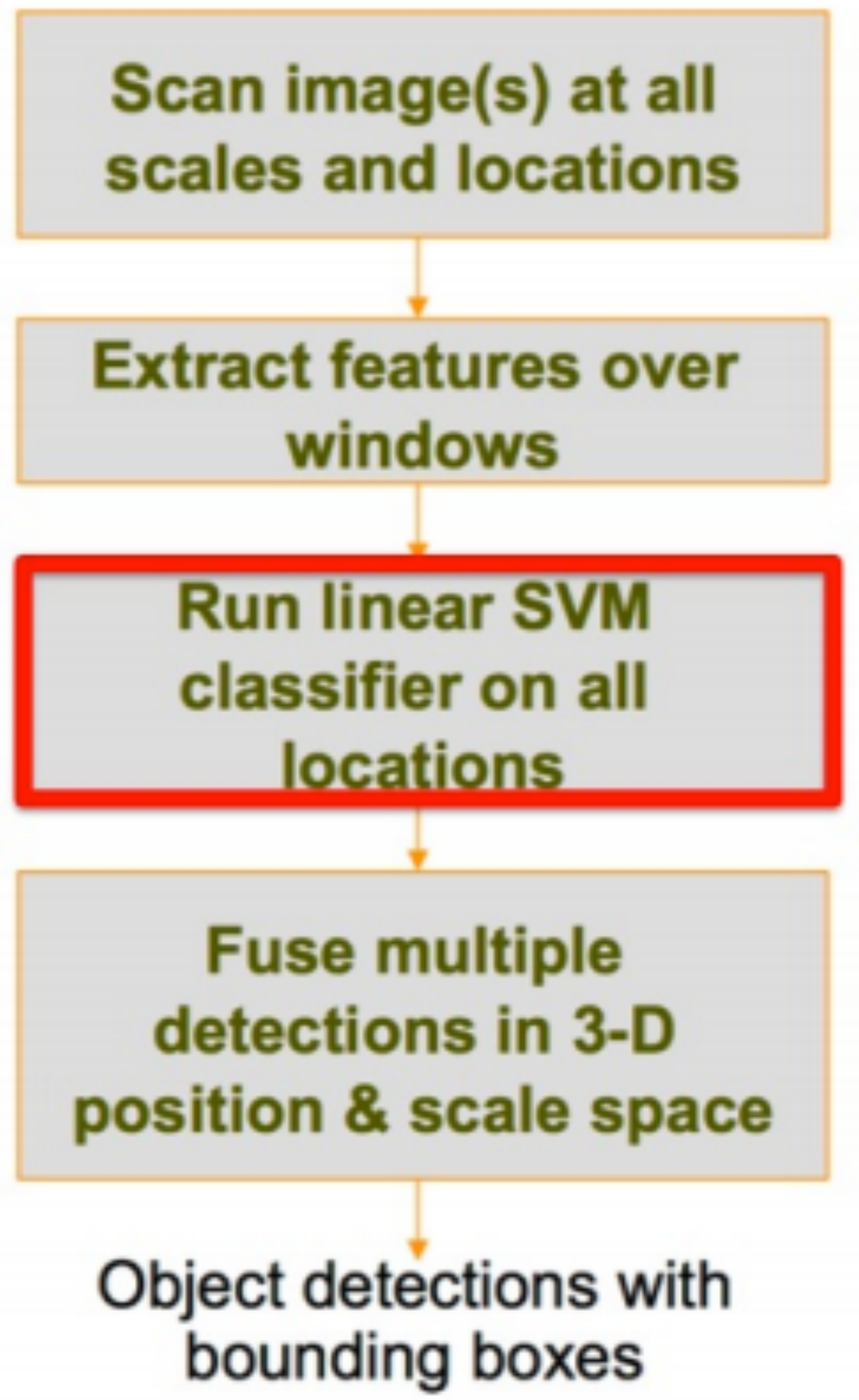
L2 normalization in each block:

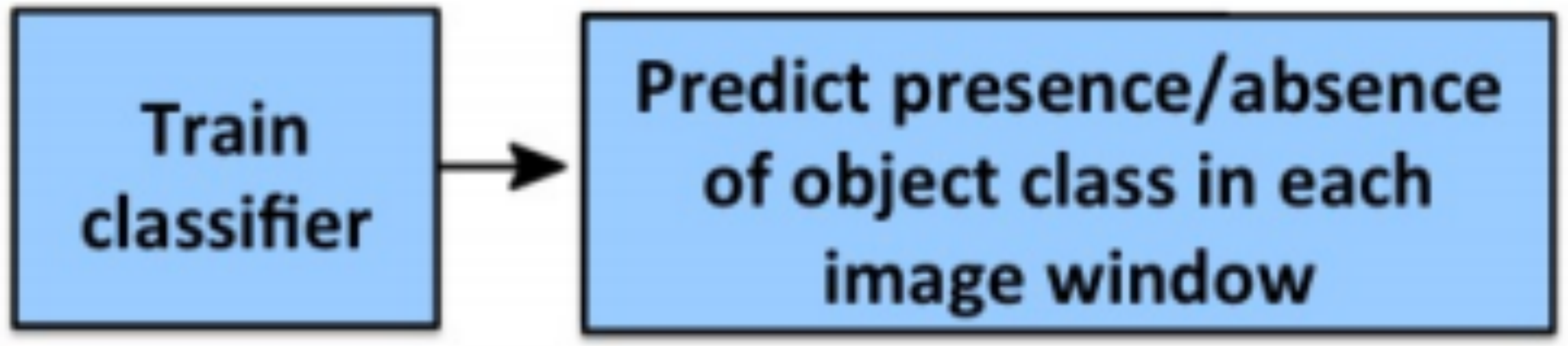
$$\mathbf{f} = \frac{\mathbf{f}}{\sqrt{||\mathbf{f}||_2^2 + \epsilon^2}}$$



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

Pics: S. Lazebnik

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!)

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

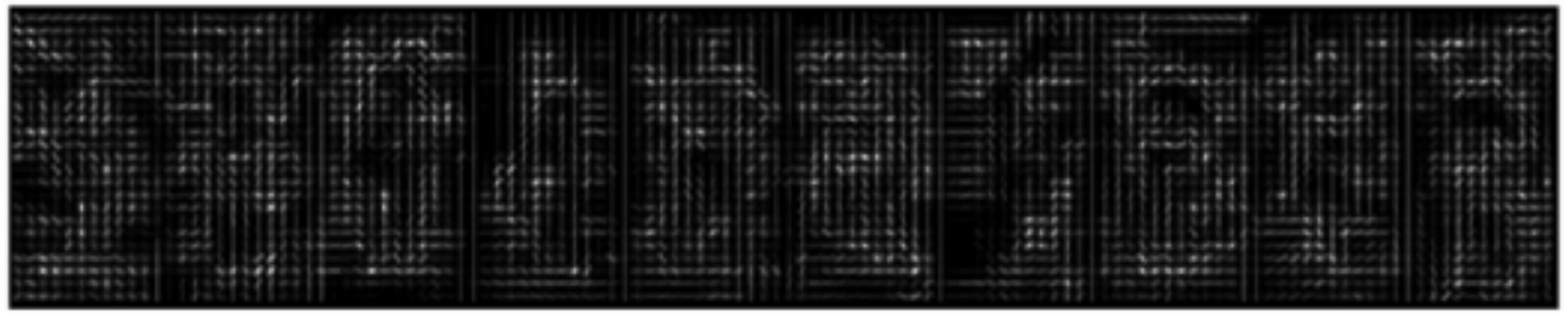
↓

Encode images into feature spaces

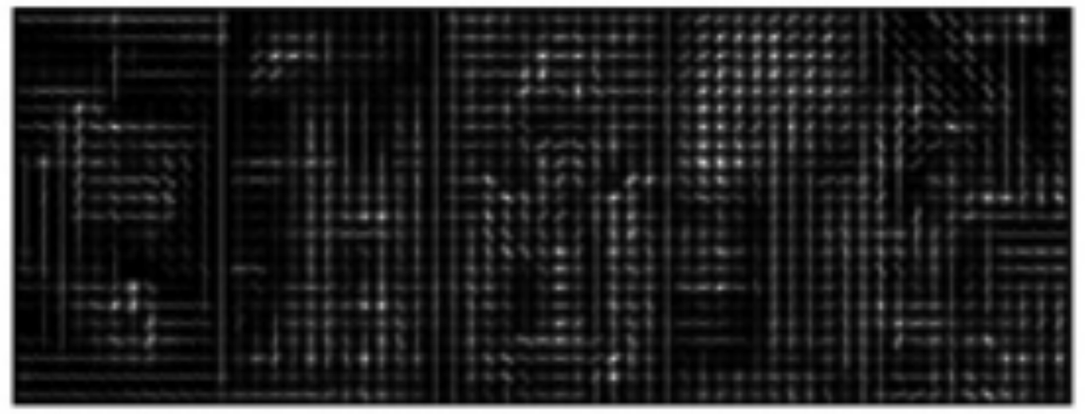
↓

Learn binary classifier

positive training examples



negative training examples

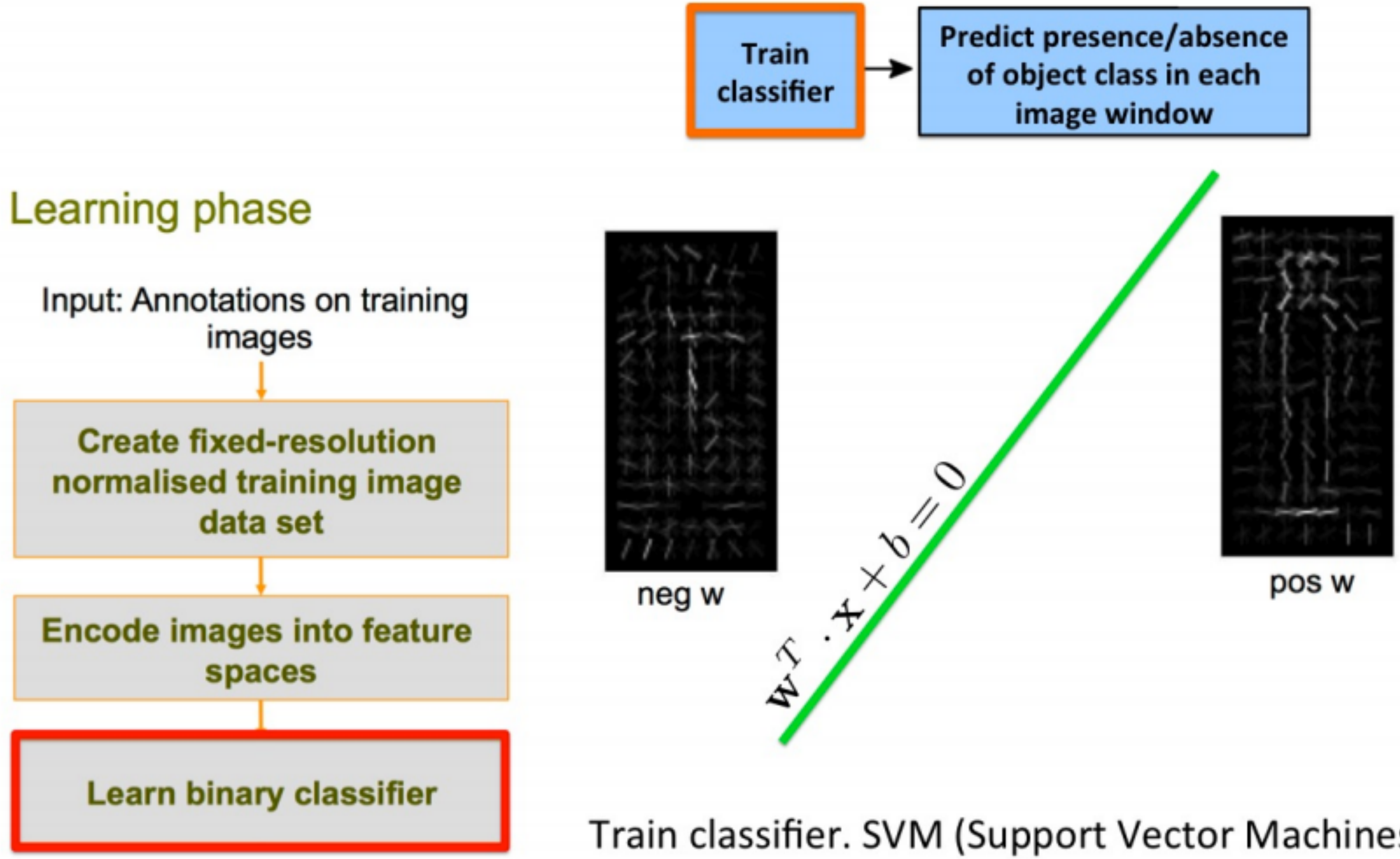


*** These are just feature visualizations. Each ``picture'' is really a 15x7x31 feature matrix.

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

Slide credit: Sanja Fidler

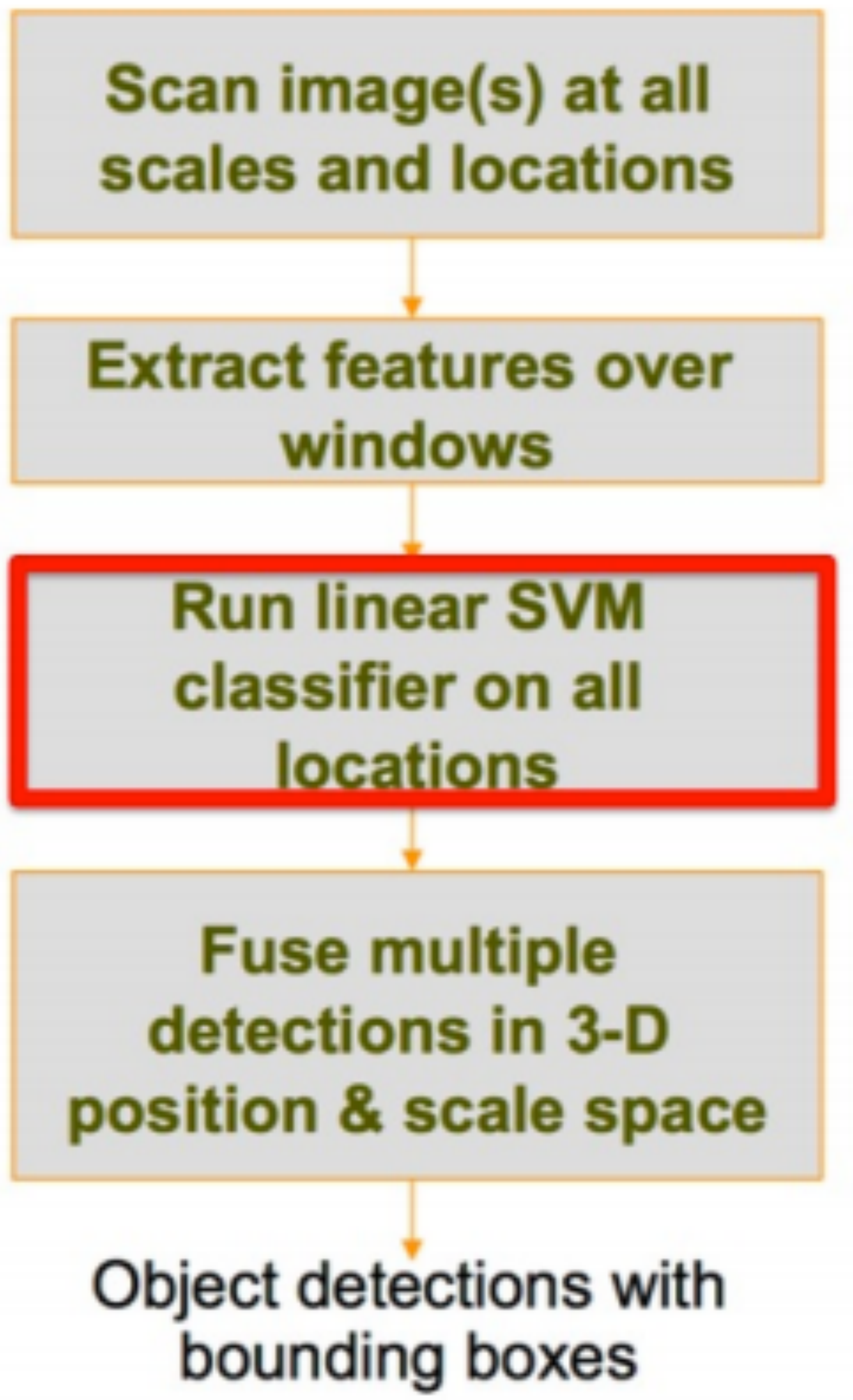
III. SVM classifier - training



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

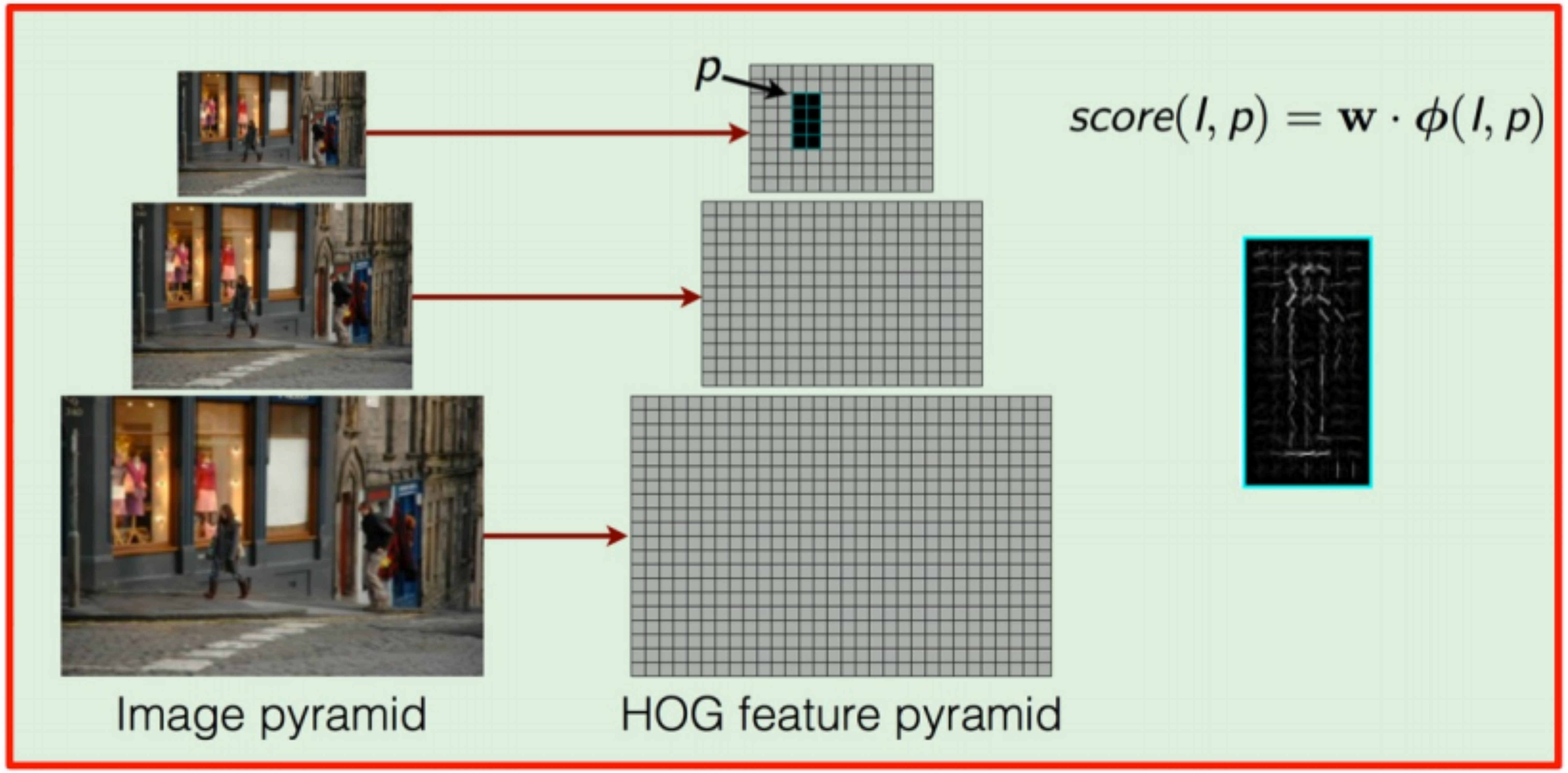
Slide credit: Sanja Fidler

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).

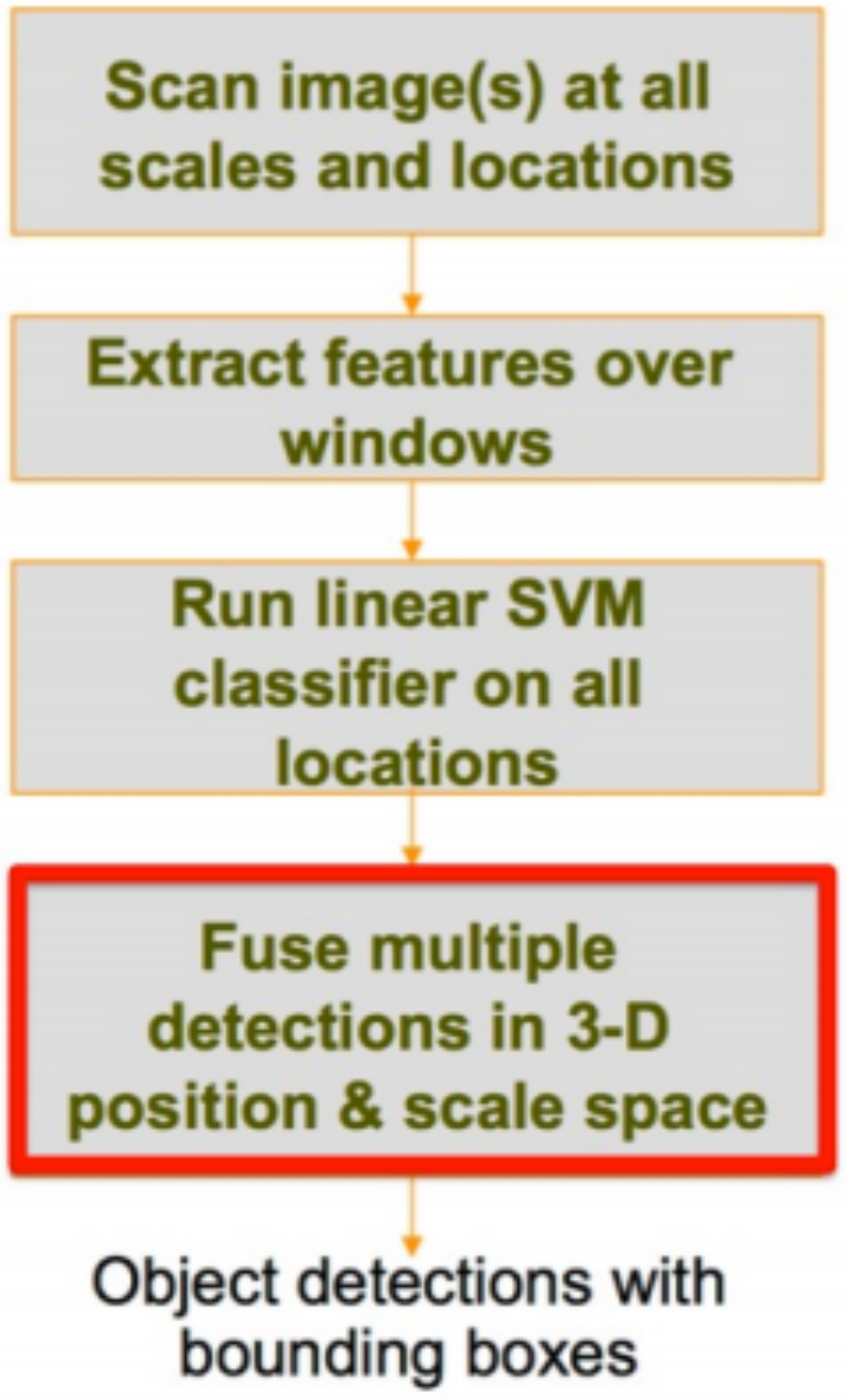
Detection Phase



[Pic from: R. Girshik]

Slide credit: Sanja Fidler

IV. Non-Maxima Suppression (NMS)



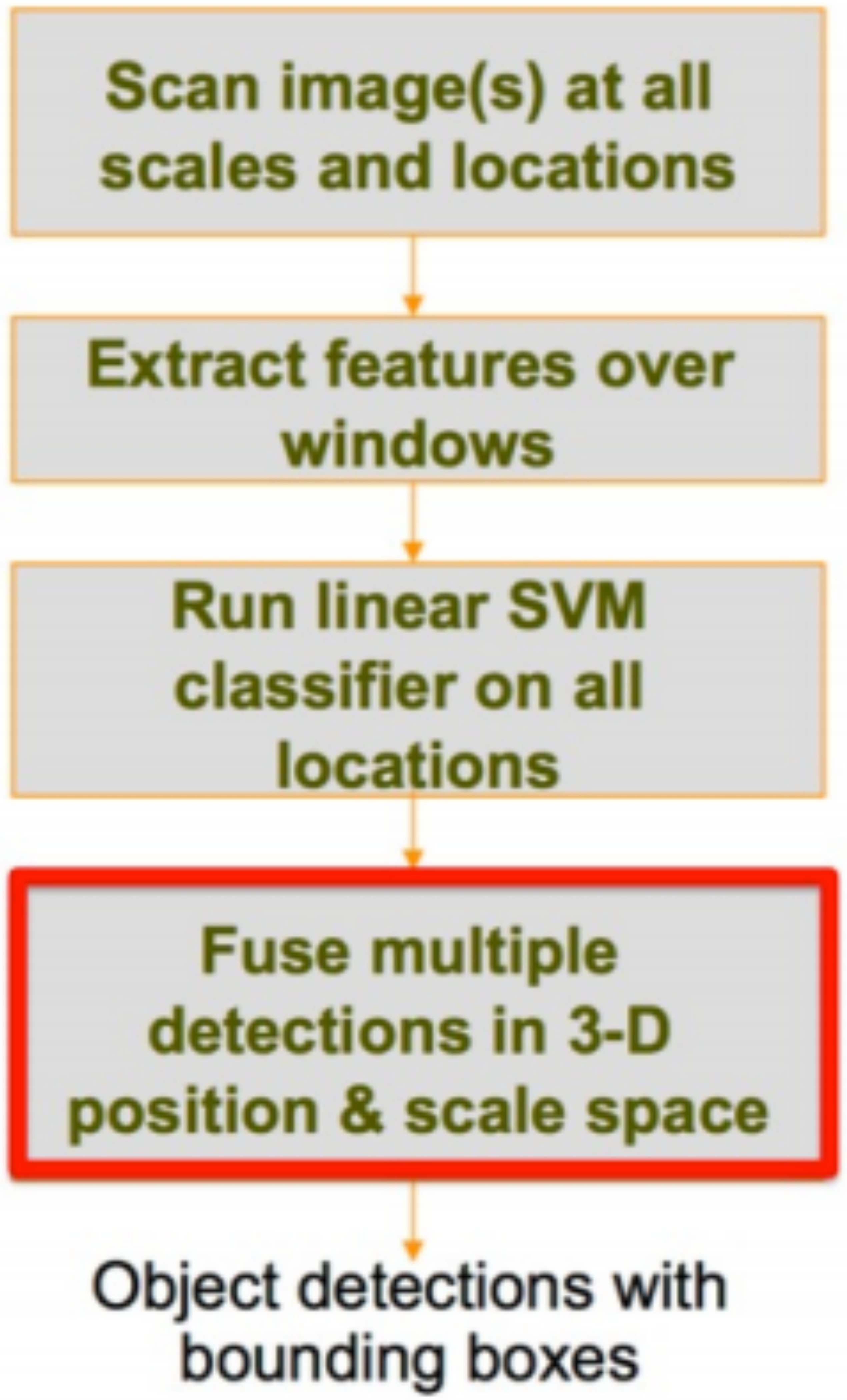
Non-maxima suppression (NMS)

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

- 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

Example: 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

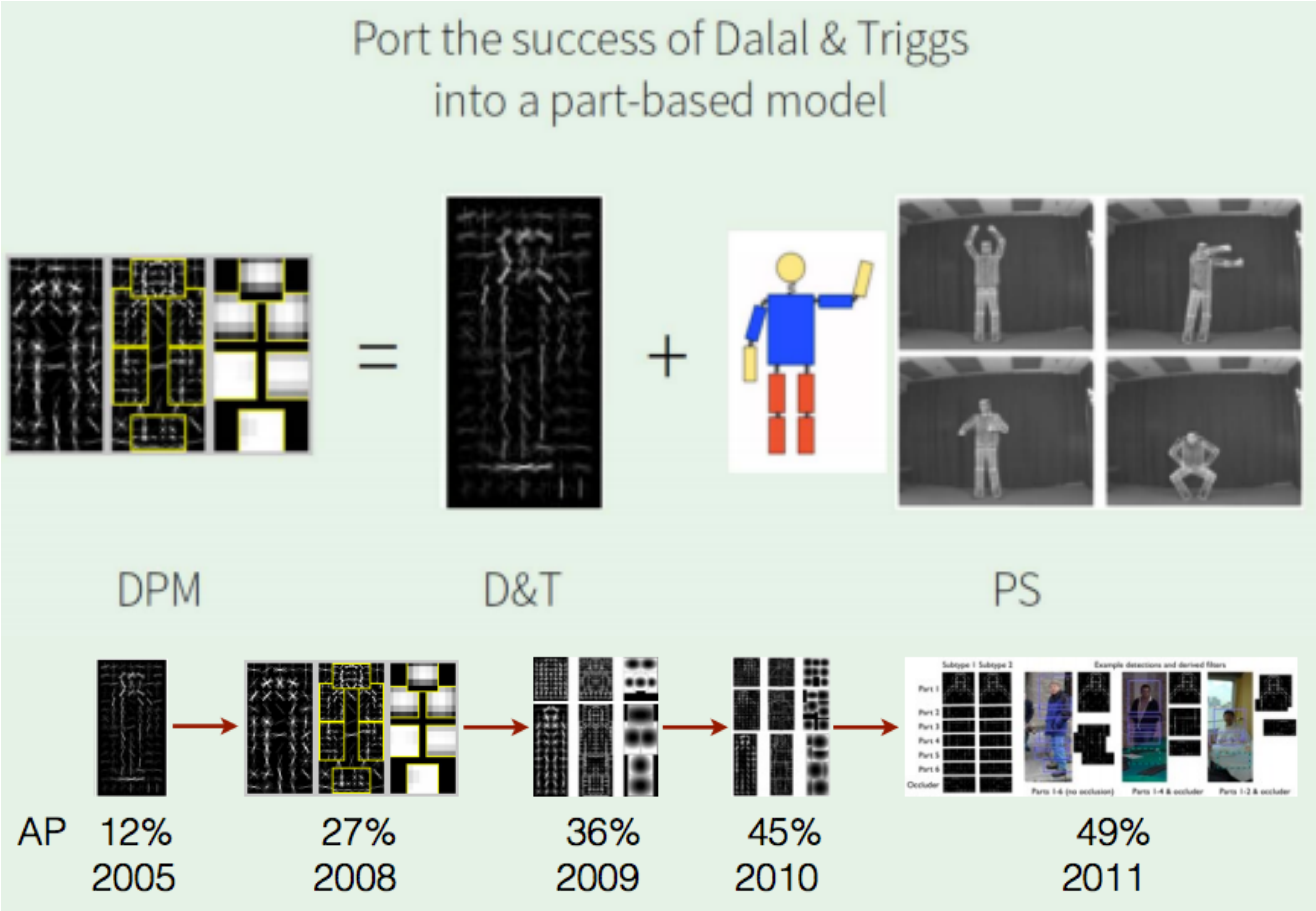
T-PAMI, 2010

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

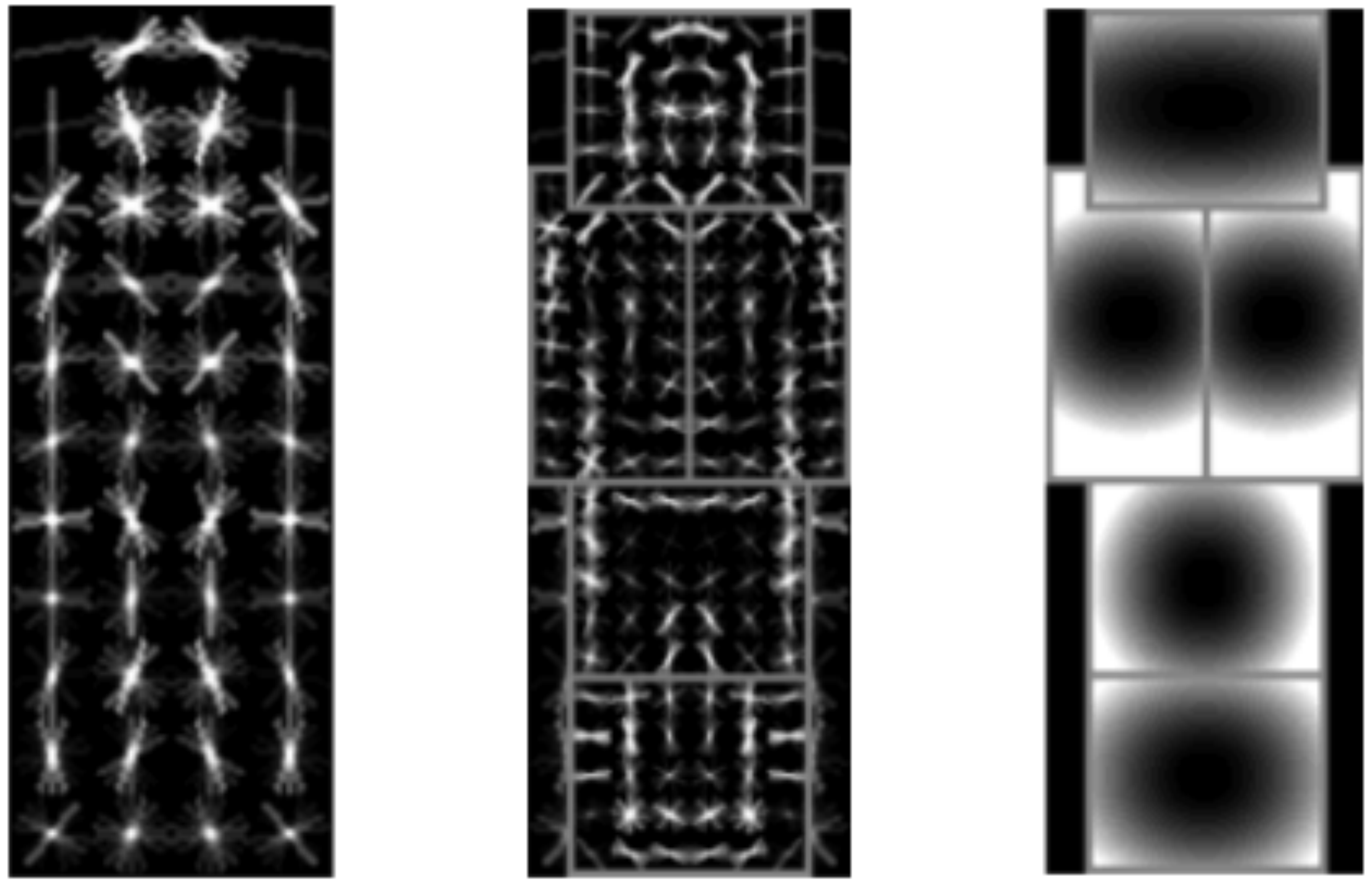
Deformable Part Model (DPM): key idea



Slide credit: Ross Girshick

DPM: Model representation

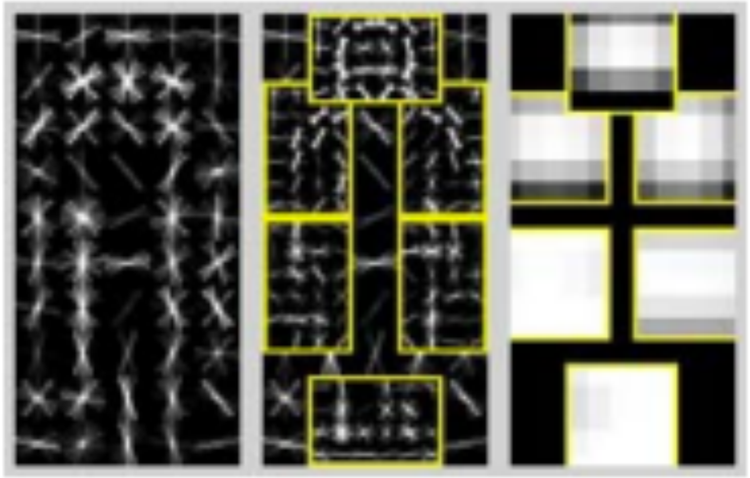
- A model has a root filter F_0 and n part models (F_i, v_i, d_i)
  - F_i: i-th part filter
  - v_i: anchor position of i-th part relative to the root
  - d_i: deformation parameters for i-th part

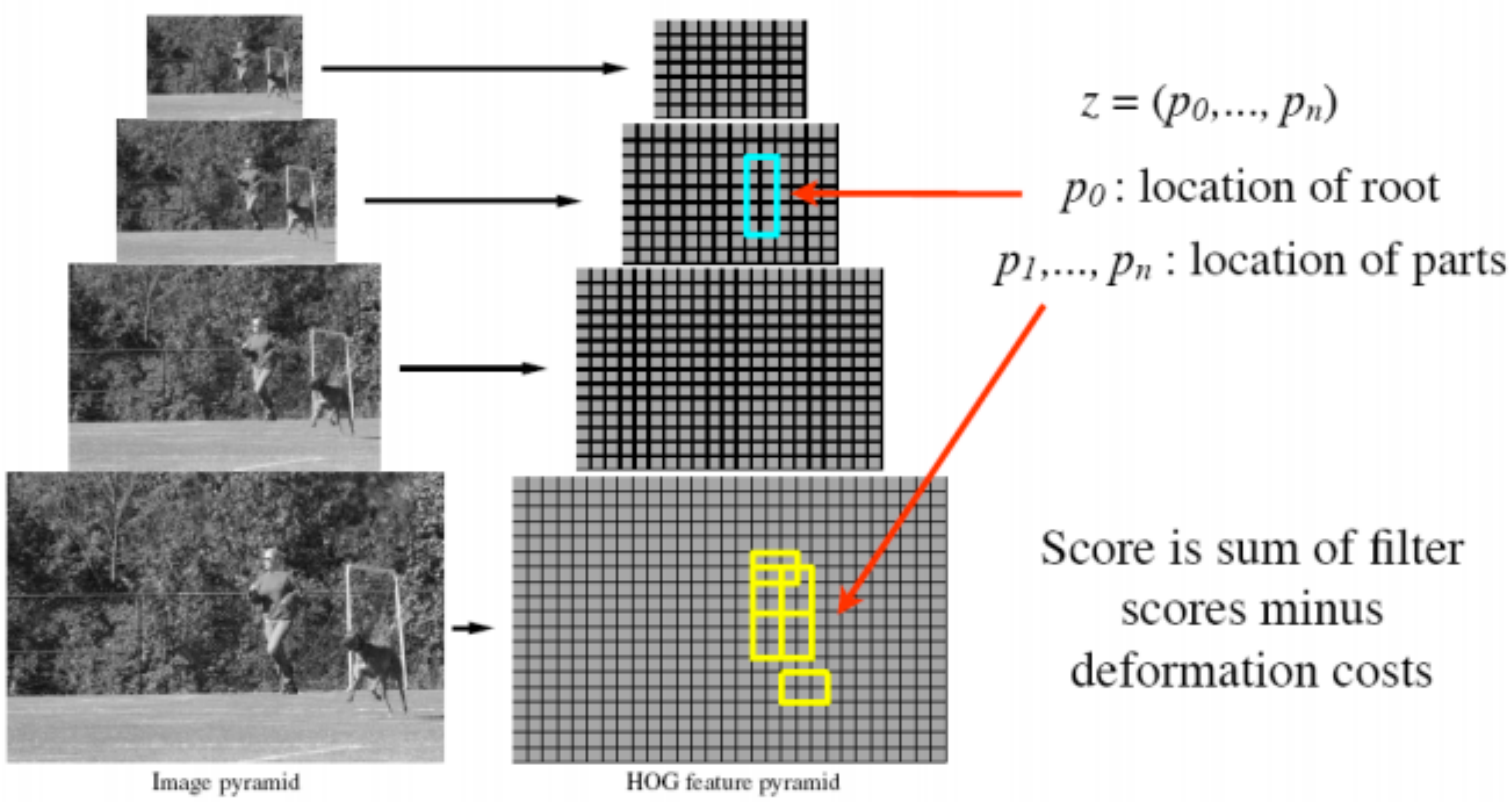


Coarse root filter — Higher resolution part filters — Deformation models

DPM: Object Hypothesis

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





Slide credit: Mubarak Shah

DPM: Score of a Hypothesis

$$score(p_0, \ldots, p_n) = \underbrace{\boxed{\sum_{i=0}^{n} \underbrace{F_i}_{\text{Filters}} \cdot \underbrace{\phi(H, p_i)}_{\text{Feature of subwindow at location } p_i}}}_{\text{data term}} - \underbrace{\boxed{\sum_{i=1}^{n} \underbrace{d_i}_{\text{Deformation parameters}} \cdot \underbrace{\phi_d(dx_i, dy_i)}_{\text{Displacement of part i relative to its anchor position}}}}_{\text{spatial prior}} + \underbrace{b}_{\text{Bias}}$$

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

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

Bias

Filters

Feature of subwindow at location p_i

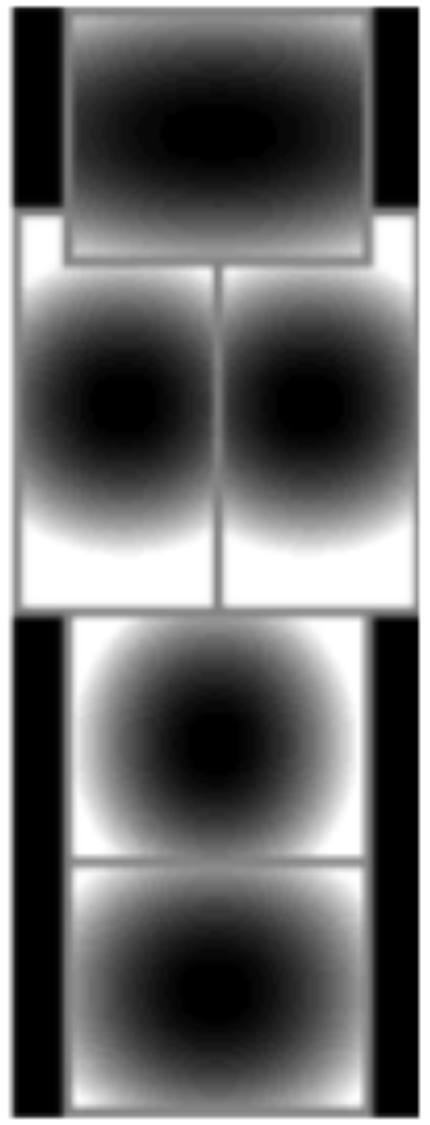
Deformation parameters

Displacement of part i relative to its anchor position

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

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

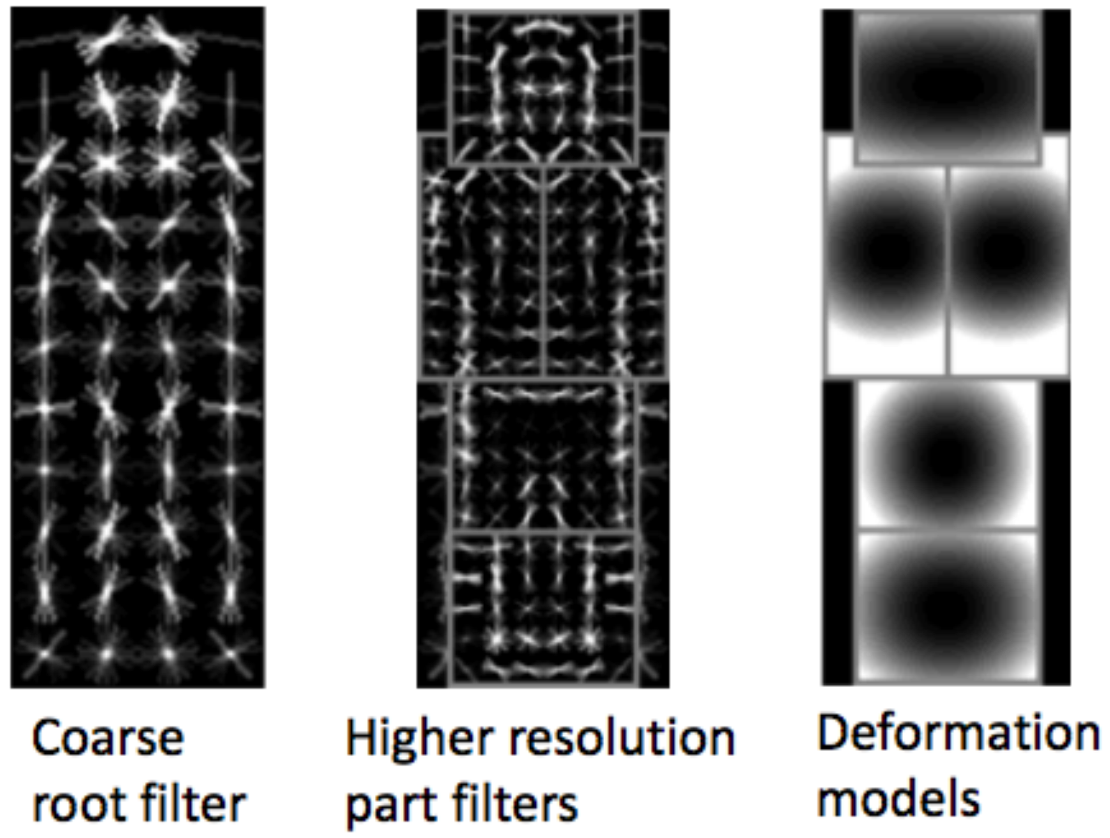
Initial Value (to be learned)



Deformation models

Slide credit: Mubarak Shah

DPM: Detection



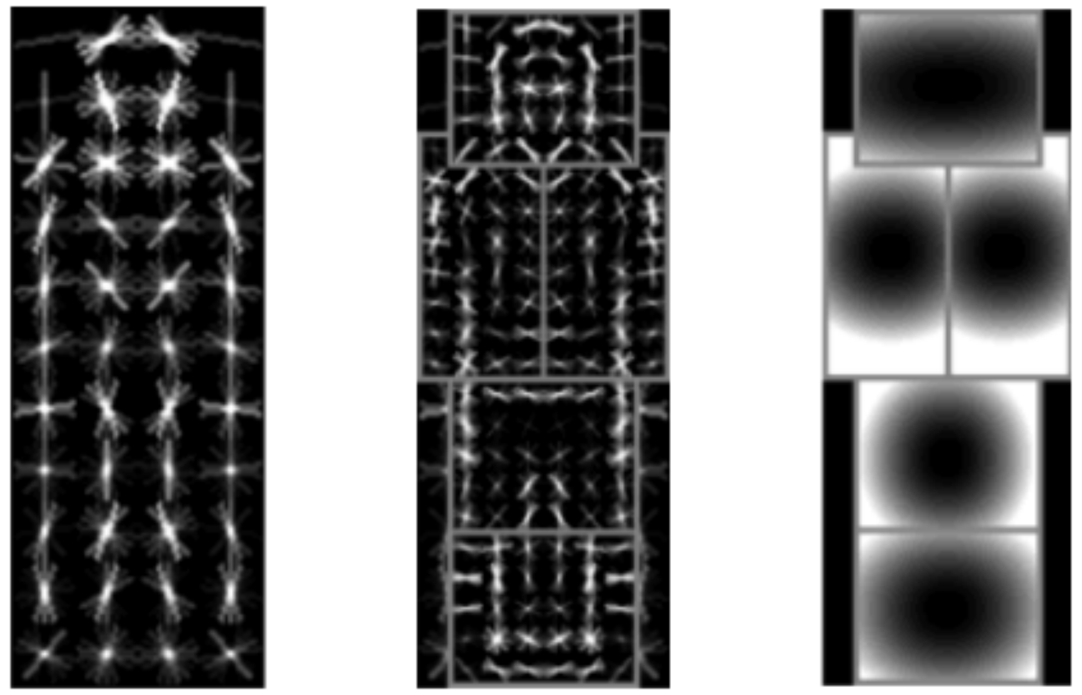
- The overall score of a root location is computed according to the best possible placement of the parts

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

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

Slide credit: Mubarak Shah

DPM: Detection



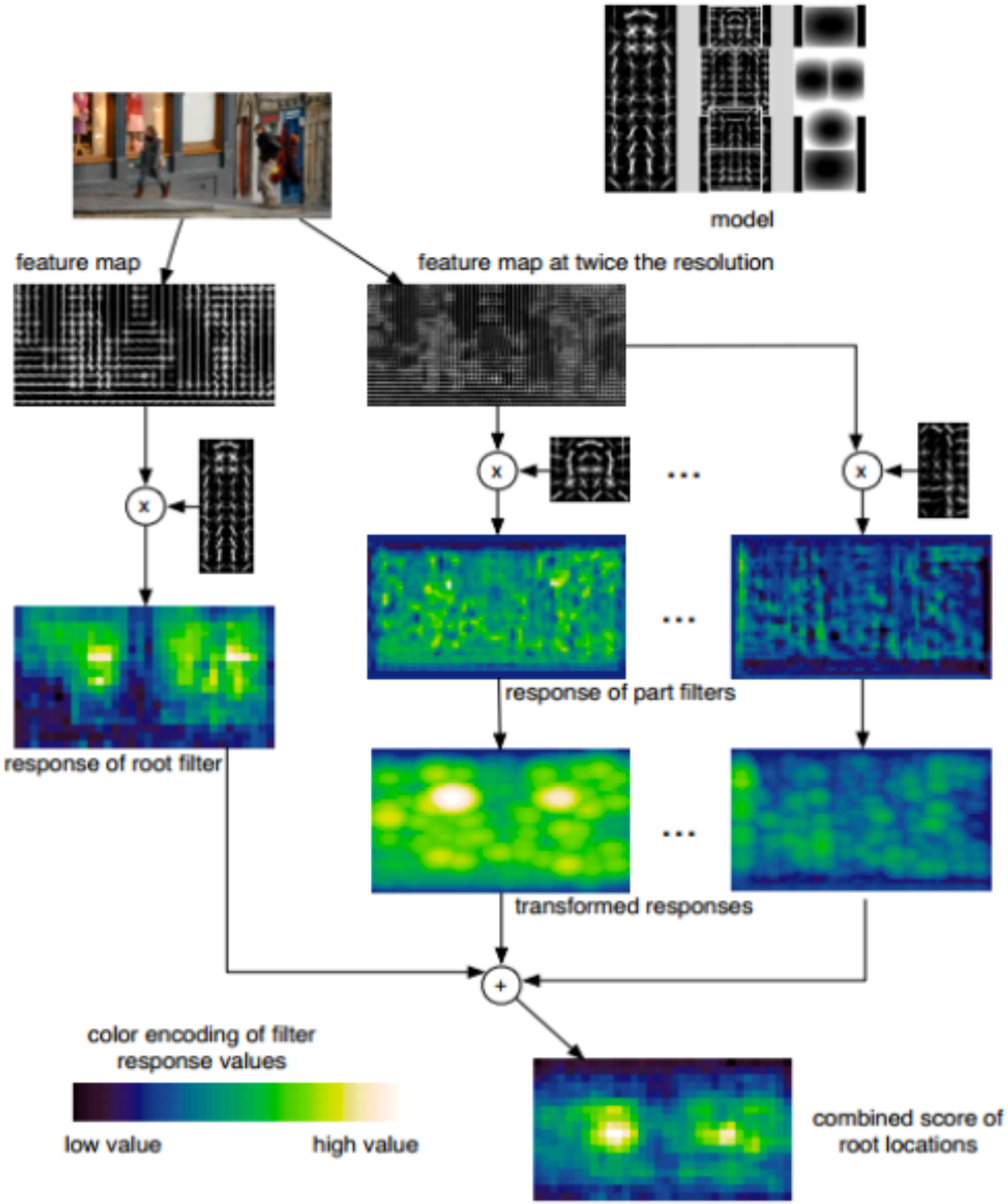
- Distance transform
  - Response of the i-th part filter in the l-th level of the feature pyramid
    $$R_{i,l}(x,y) = F_i \cdot \phi(H,(x,y,l))$$
  - Transformed response, given root is at *(x,y)*
    $$D_{i,l}(x,y) = \max_{dx,dy}(R_{i,l}(x+dx,y+dy) - d_i \cdot \phi_d(dx,dy))$$

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

(x,y)

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

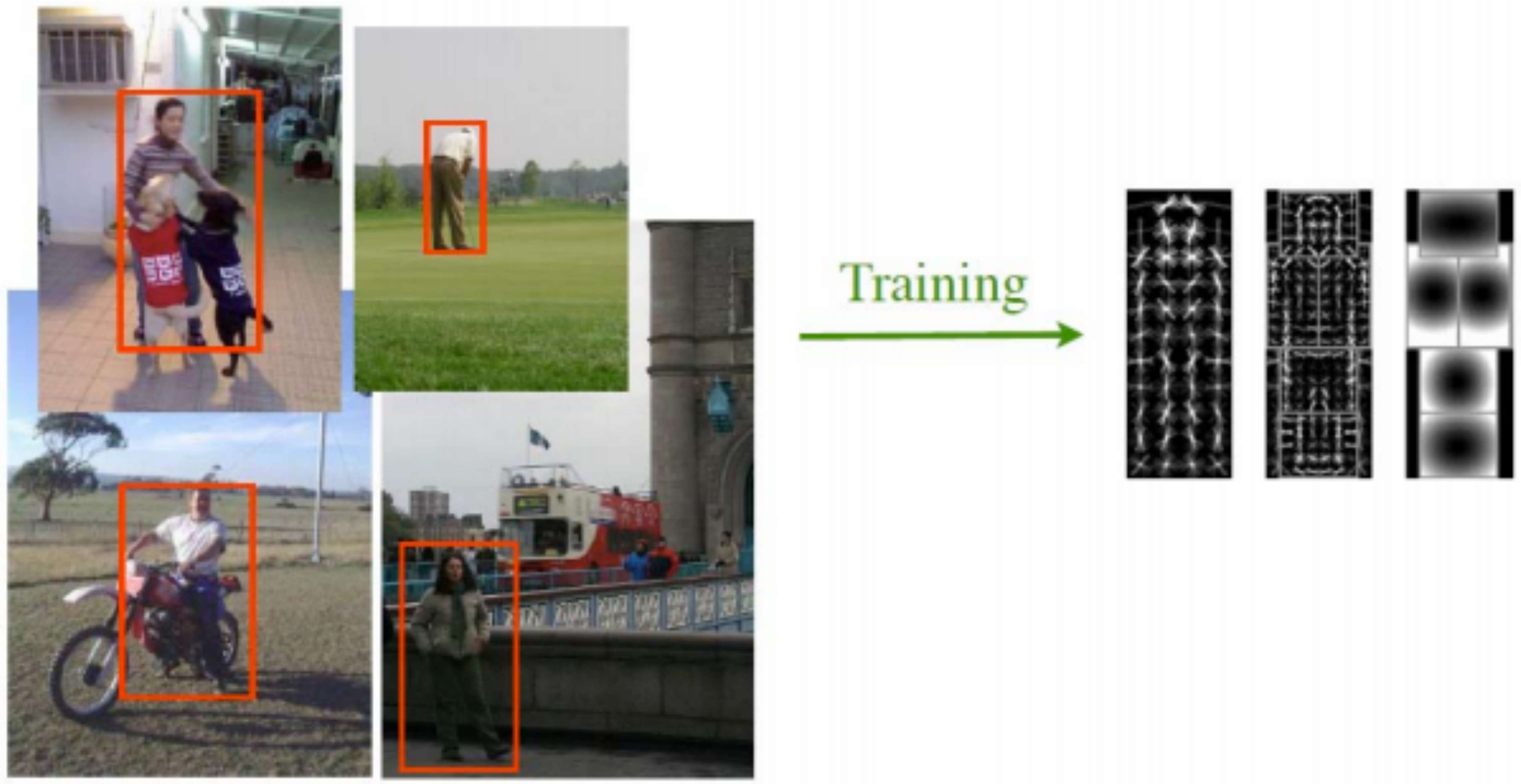
DPM: Detection



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

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

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

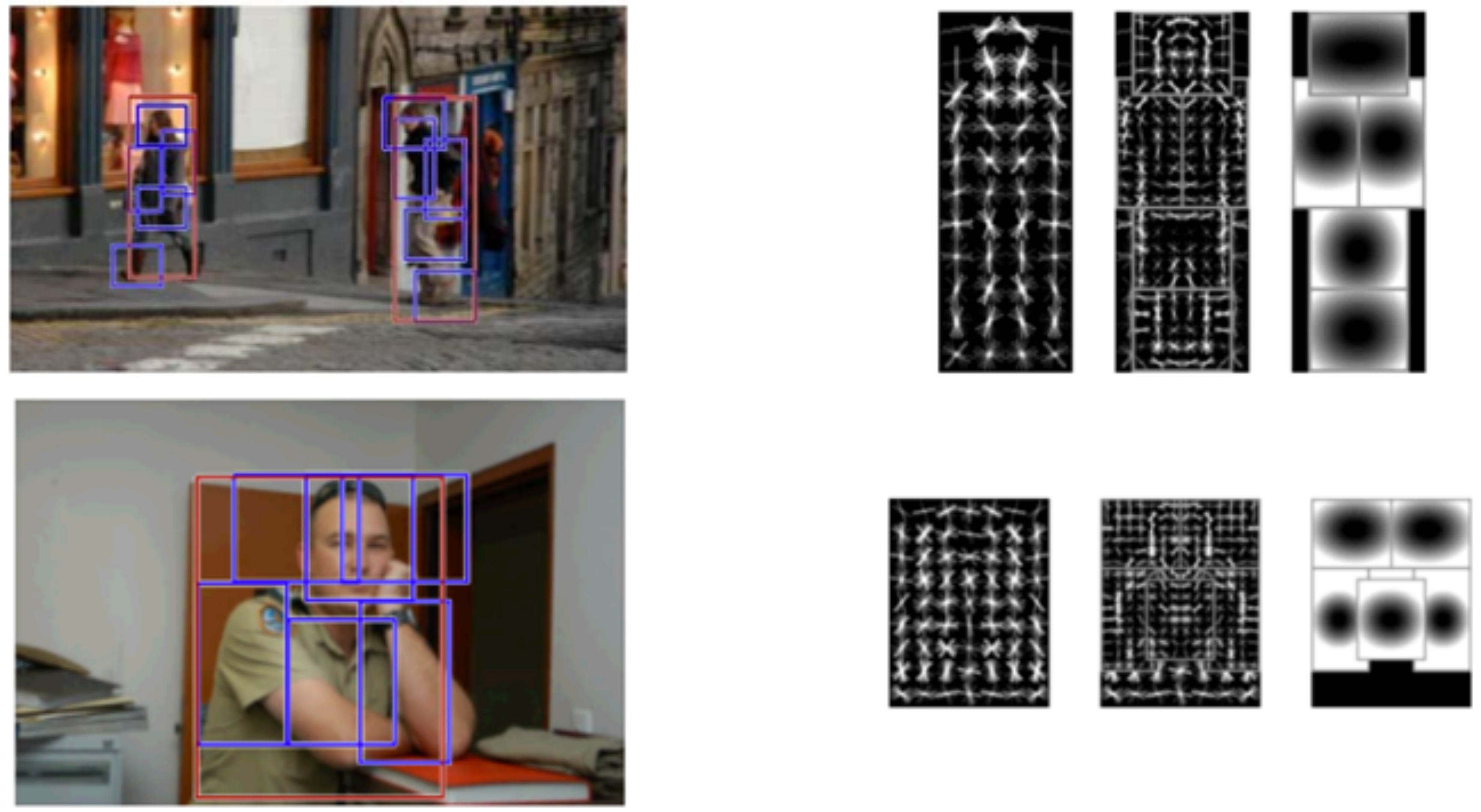
- Initialize β using standard SVM by assuming the same parts locations for all the positive examples
- Iterative optimization:
  - Relabel positive examples: fix β, find the best z for each positive example (exactly the same with detection!)
  - Optimize β: fix z, optimize β by solving the convex problem

Slide credit: Mubarak Shah

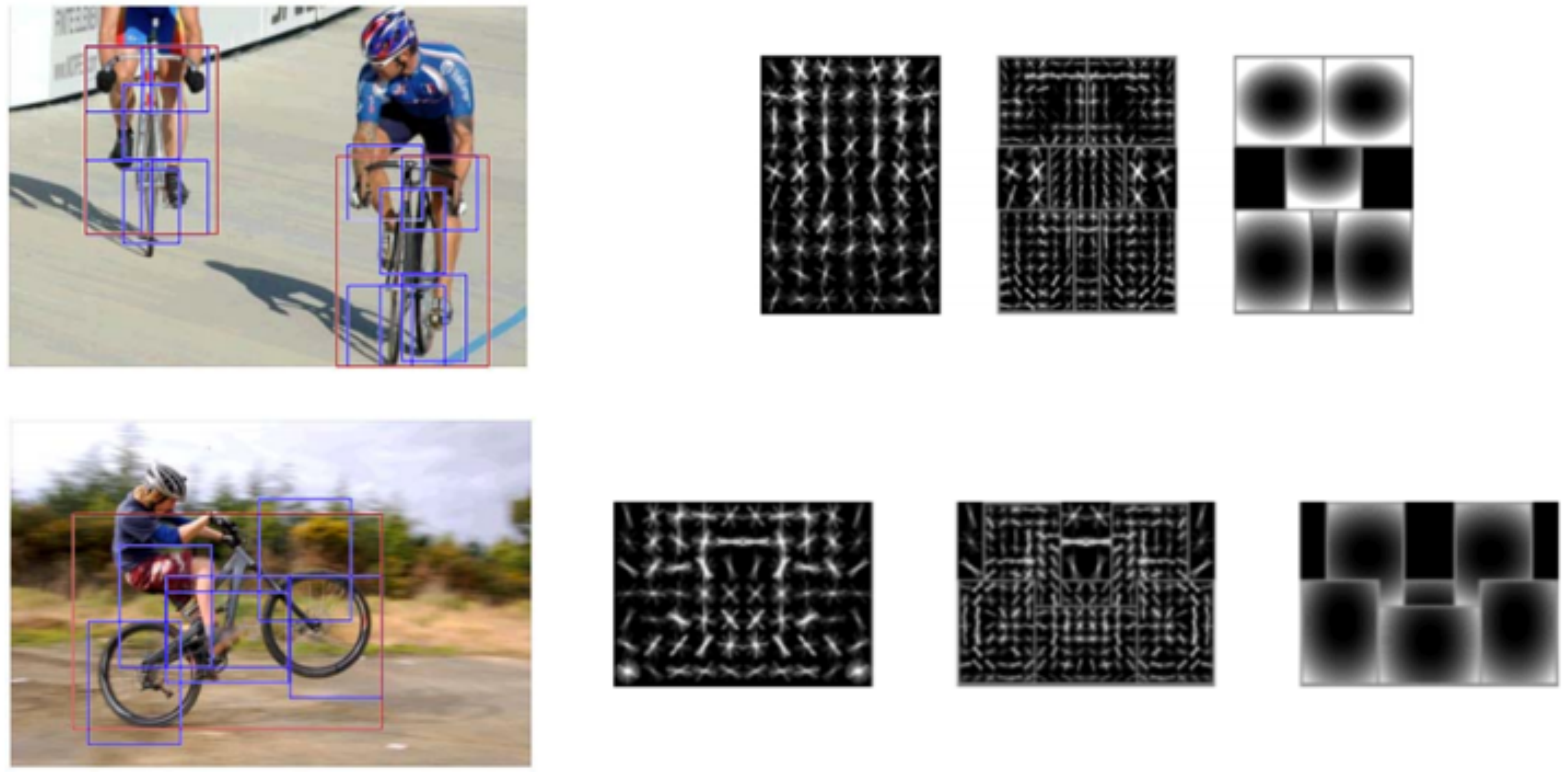
DPM: Mixture model

- 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

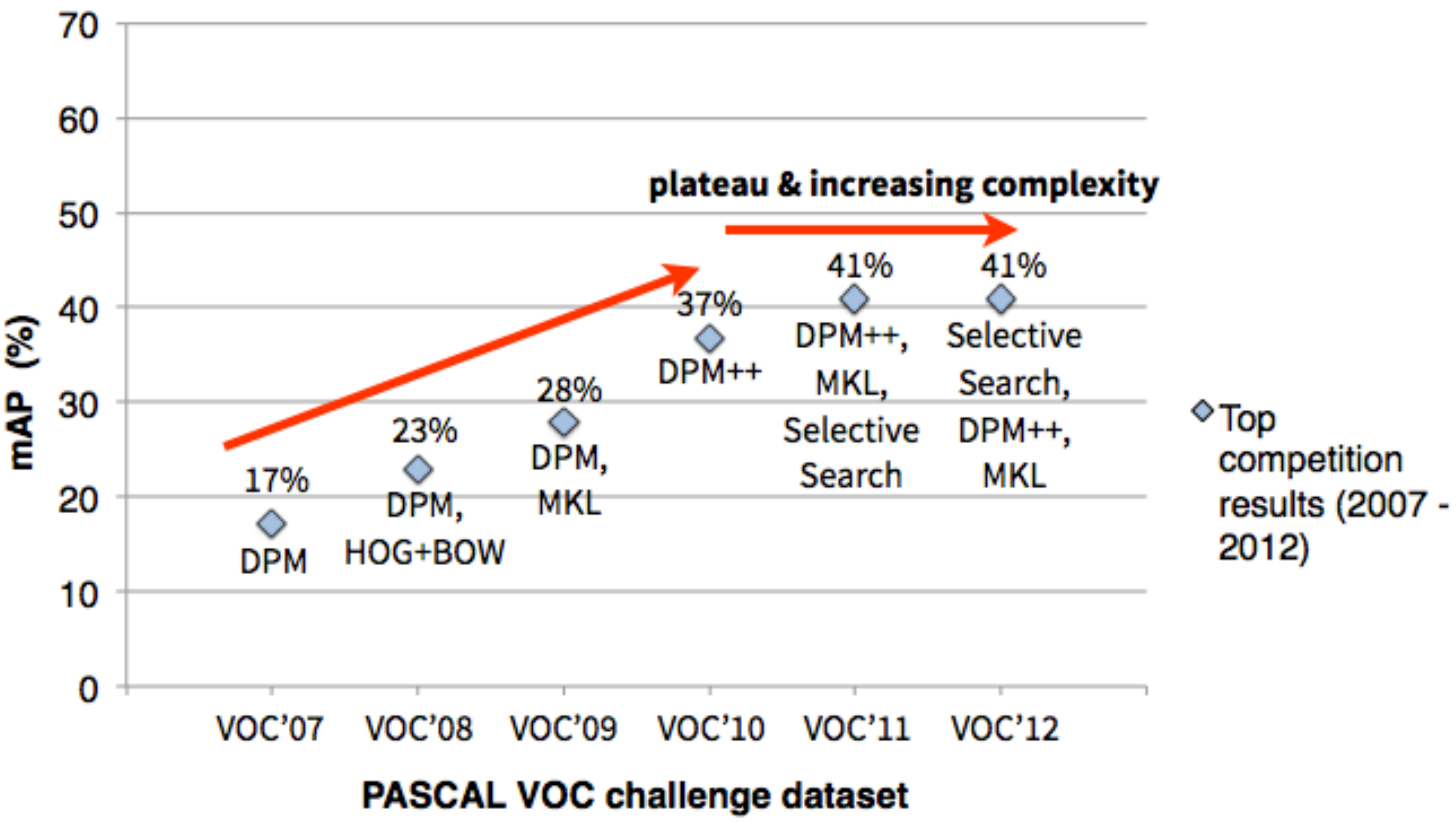


Mixture Model Example - Bicycle



Slide credit: Mubarak Shah

DPM on PASCAL VOC



Ross Girshick

Lifetime Achievement Award
by PASCAL VOC

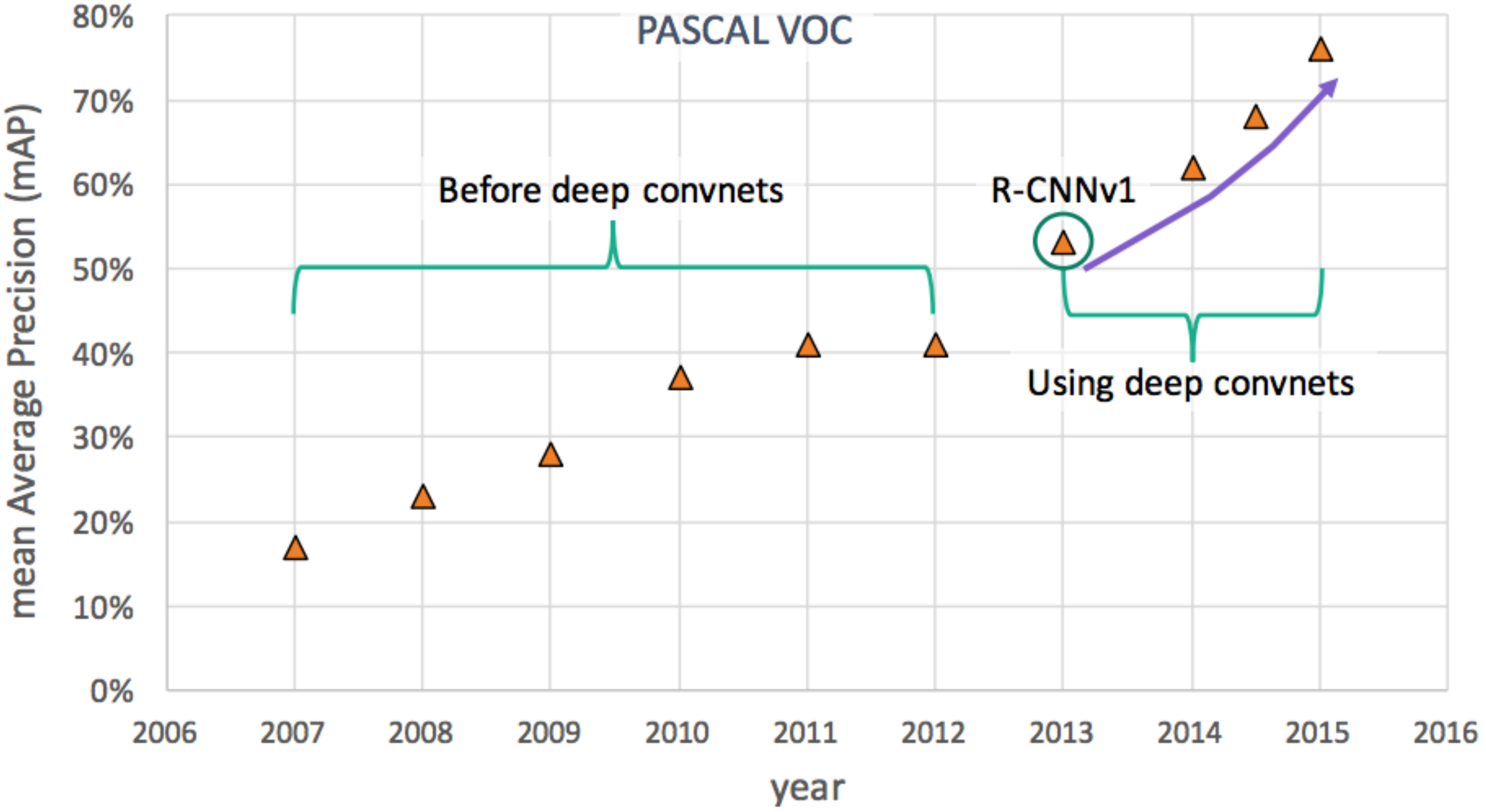
[Source: http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc20{07,08,09,10,11,12}/results/index.html]

Slide credit: Ross Girshick

Object detection

- Introduction
- Face detection: from Viola-Jones to CNN
- General object detection
  - HOG detector
  - Deformable Part-based Model
  - Region-CNN
  - Fast versions of R-CNN
  - YOLO/SSD
- Future directions

Object detection renaissance (2013-present)



Slide credit: Renjie Liao

Deep object detection

Object Detection

Published: 09 Oct 2015 Category: deep_learning

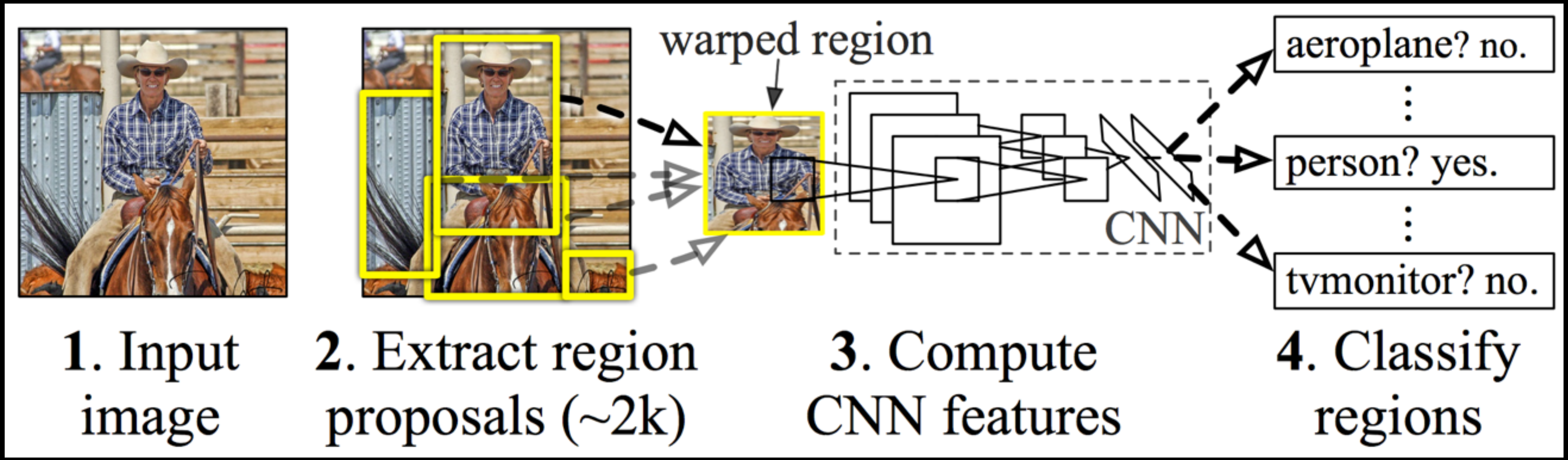
Jump to...

- 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

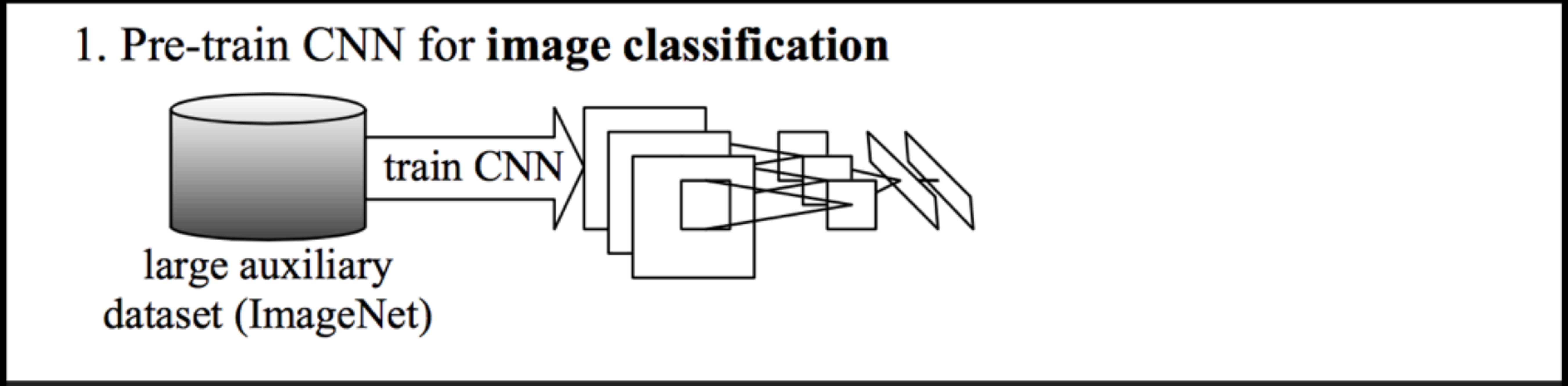
Slide credit: https://handong1587.github.io/deep_learning/2015/10/09/object-detection.html

R-CNN: Regions with CNN features



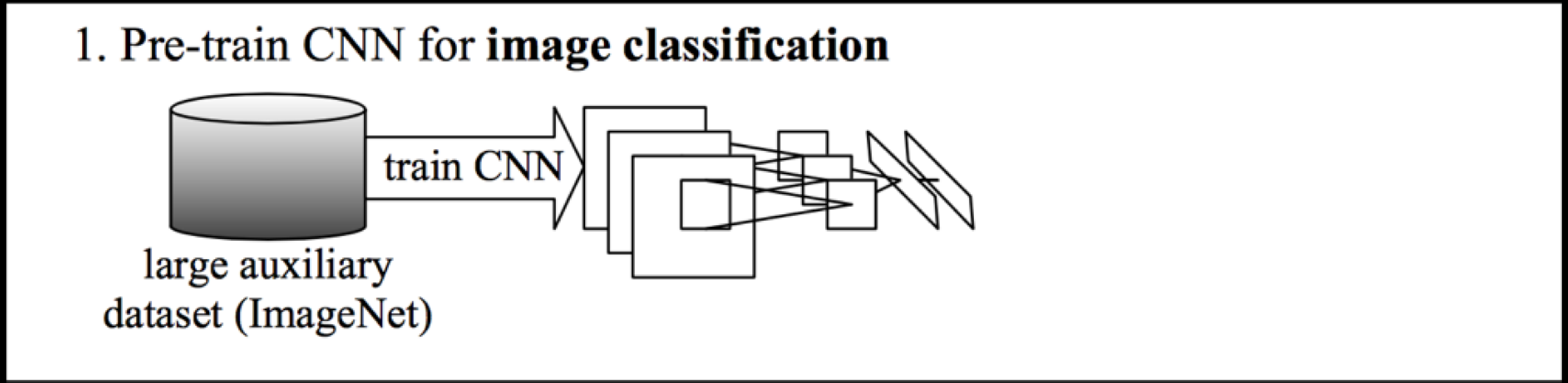
Slide credit: Girshick R, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. CVPR2014

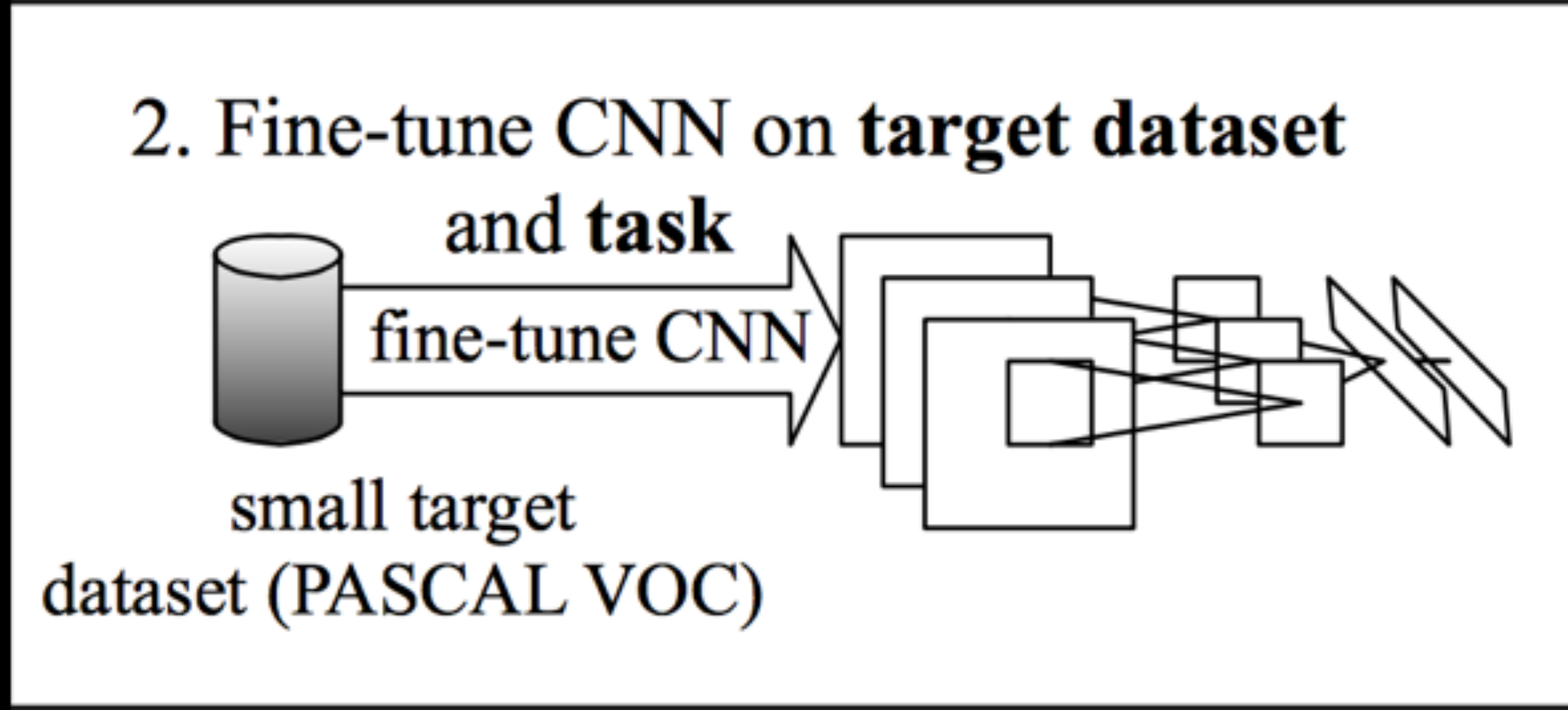
Training



Slide credit: Ross Girshick

Training

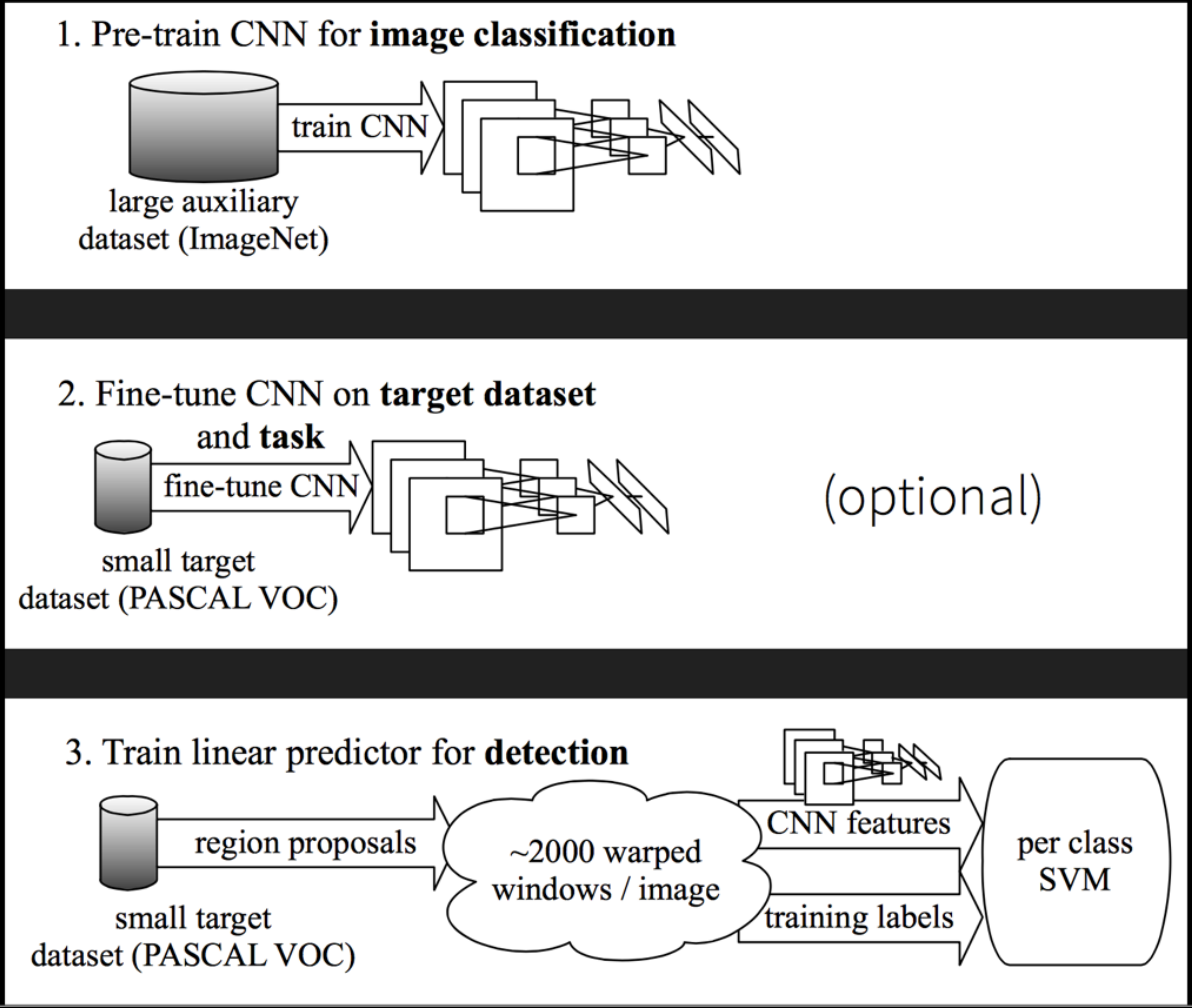




(optional)

Slide credit: Ross Girshick

Training



Slide credit: Ross Girshick

R-CNN Results

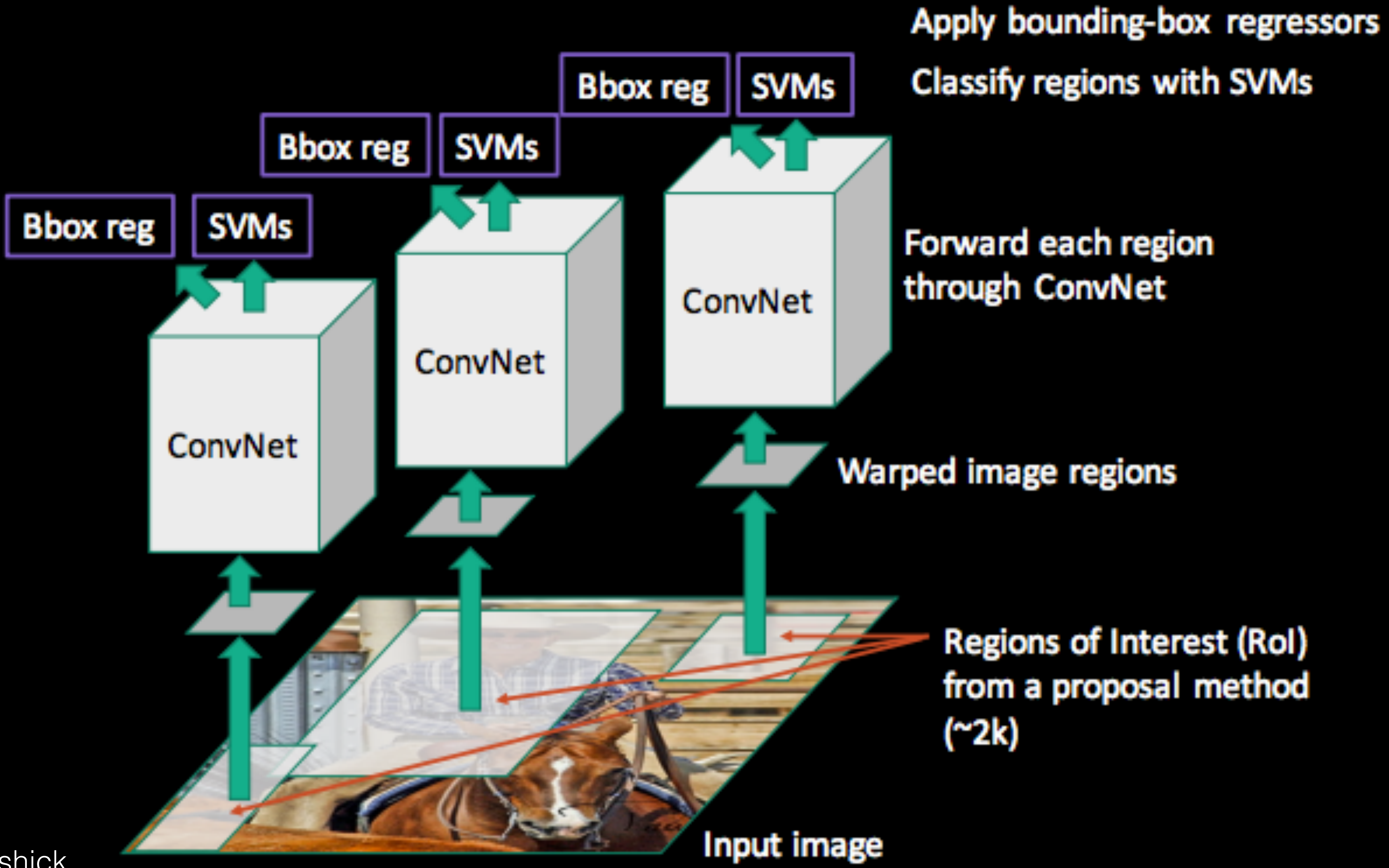
	VOC2007
DPM v5 (Girshick et al. 2011)	33.7%
Regionlets (Wang et al. 2013)	41.7%
R-CNN (AlexNet)	54.2%
R-CNN (AlexNet) + BB	58.5%
R-CNN (VGGNet)	62.2%
R-CNN (VGGNet) + BB	66.0%

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%

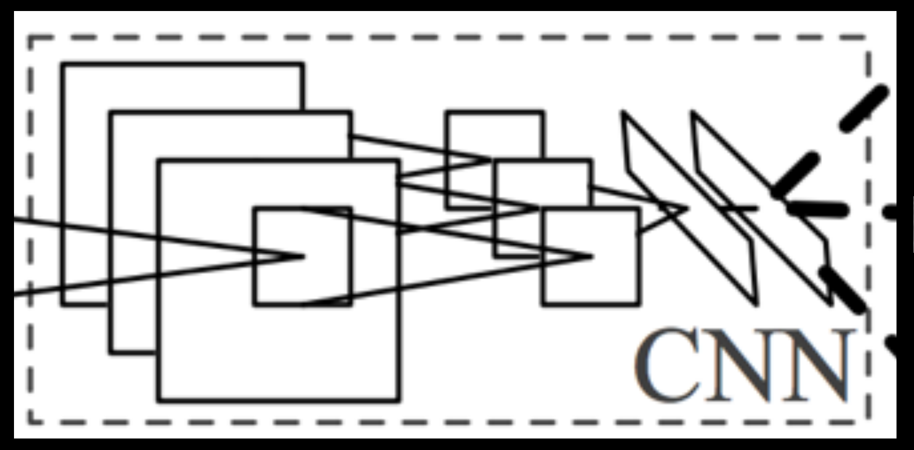
R-CNN (VGGNet)	Time
Train	84 hours
Test	47 s/im

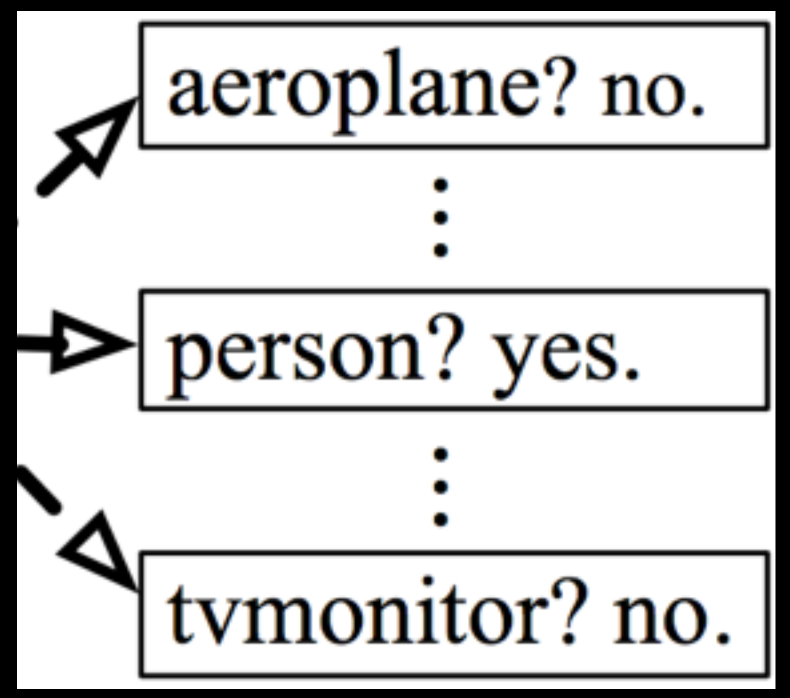
Slow R-CNN



Slide credit: Ross Girshick

Object Detection System



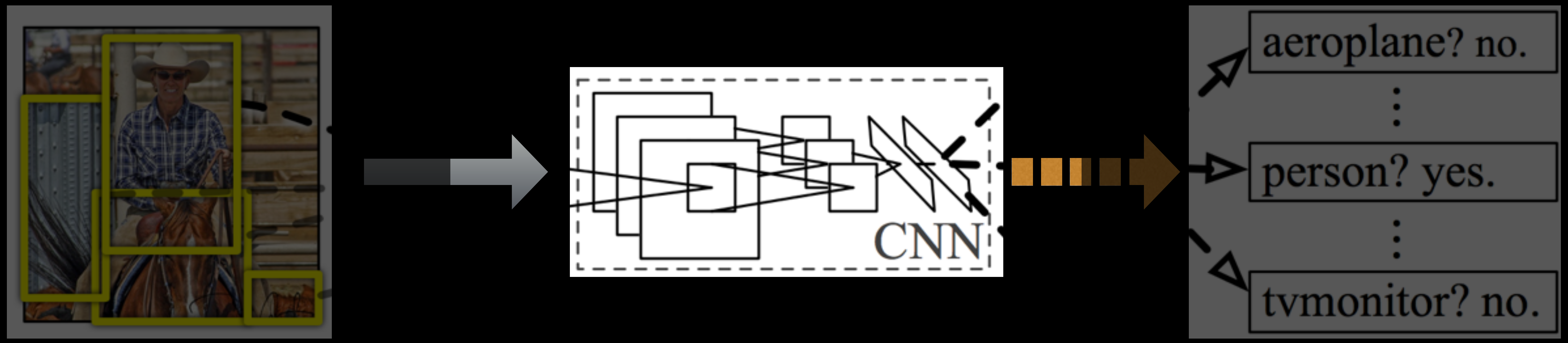


Getting Proposals

Feature Extraction

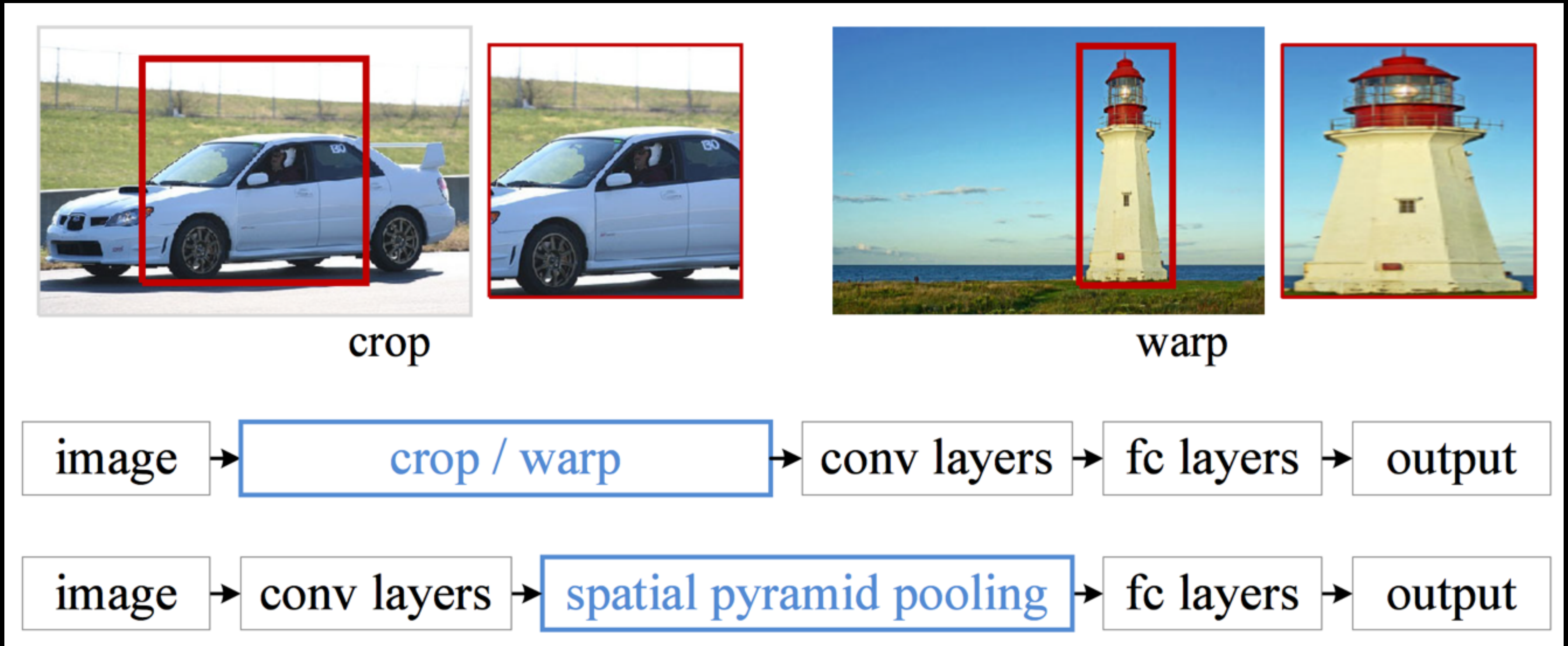
Classifier

Object Detection System



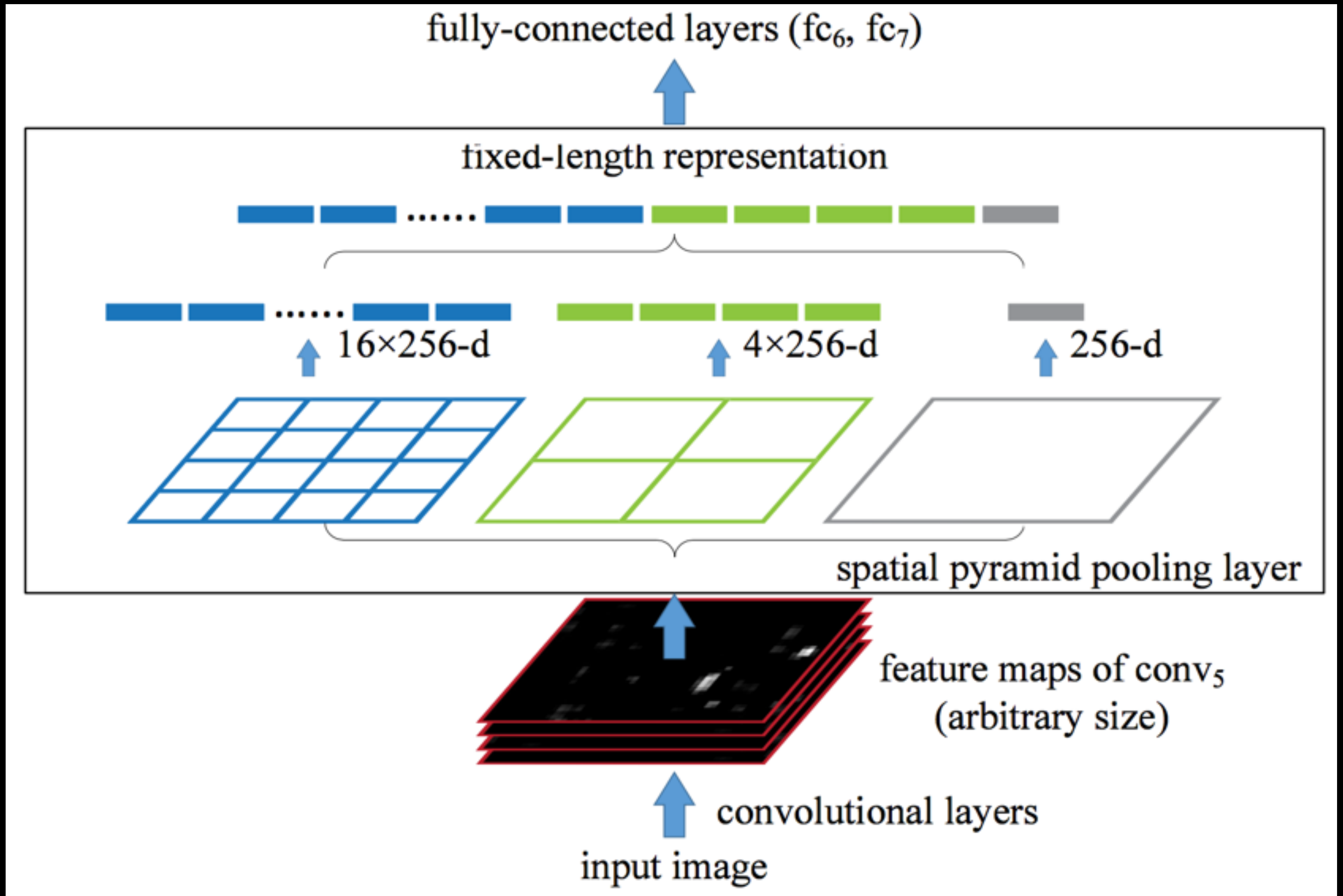
Getting Proposals **Feature Extraction** Classifier

Spatial Pyramid Pooling



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

Spatial Pyramid Pooling

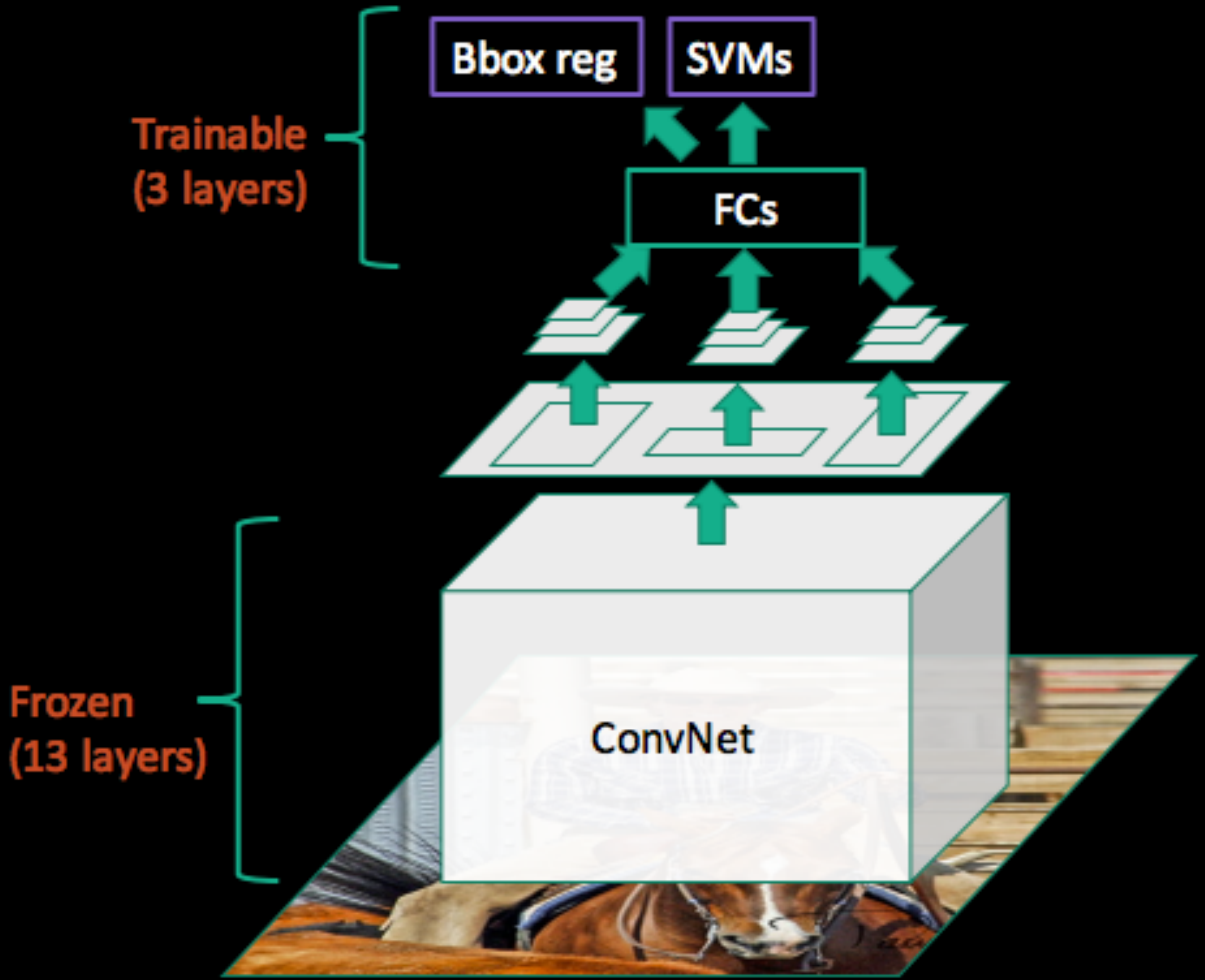



```
[pool3x3]           [pool2x2]           [pool1x1]
type=pool           type=pool           type=pool
pool=max            pool=max            pool=max
inputs=conv5        inputs=conv5        inputs=conv5
sizeX=5             sizeX=7             sizeX=13
stride=4            stride=6            stride=13

[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

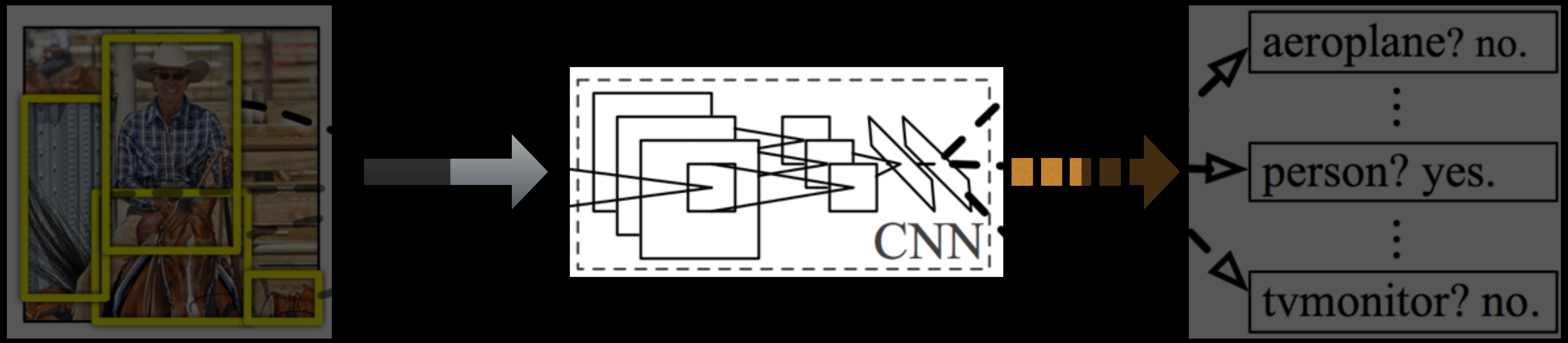


Slide credit: Ross Girshick

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

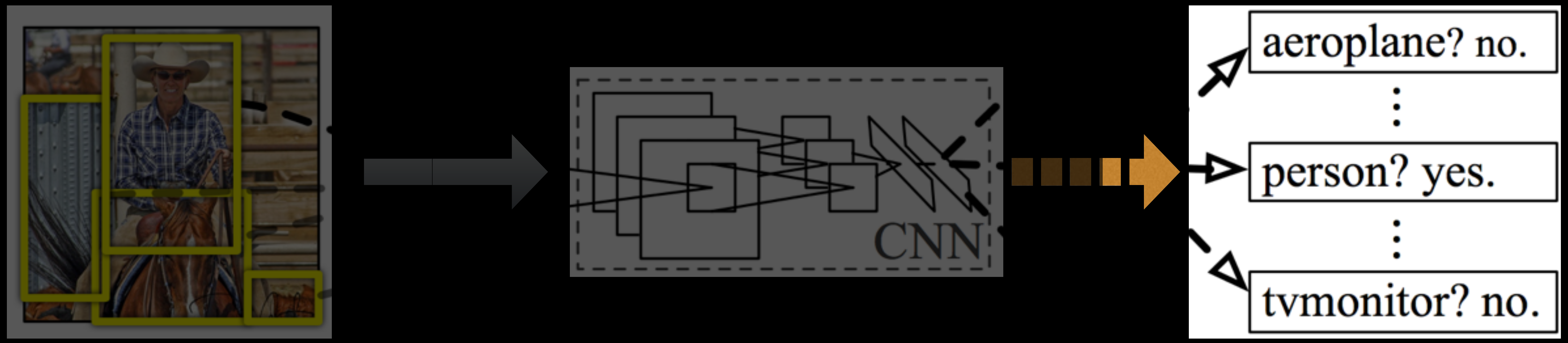
Object Detection System



Getting Proposals **Feature Extraction** Classifier

SPP

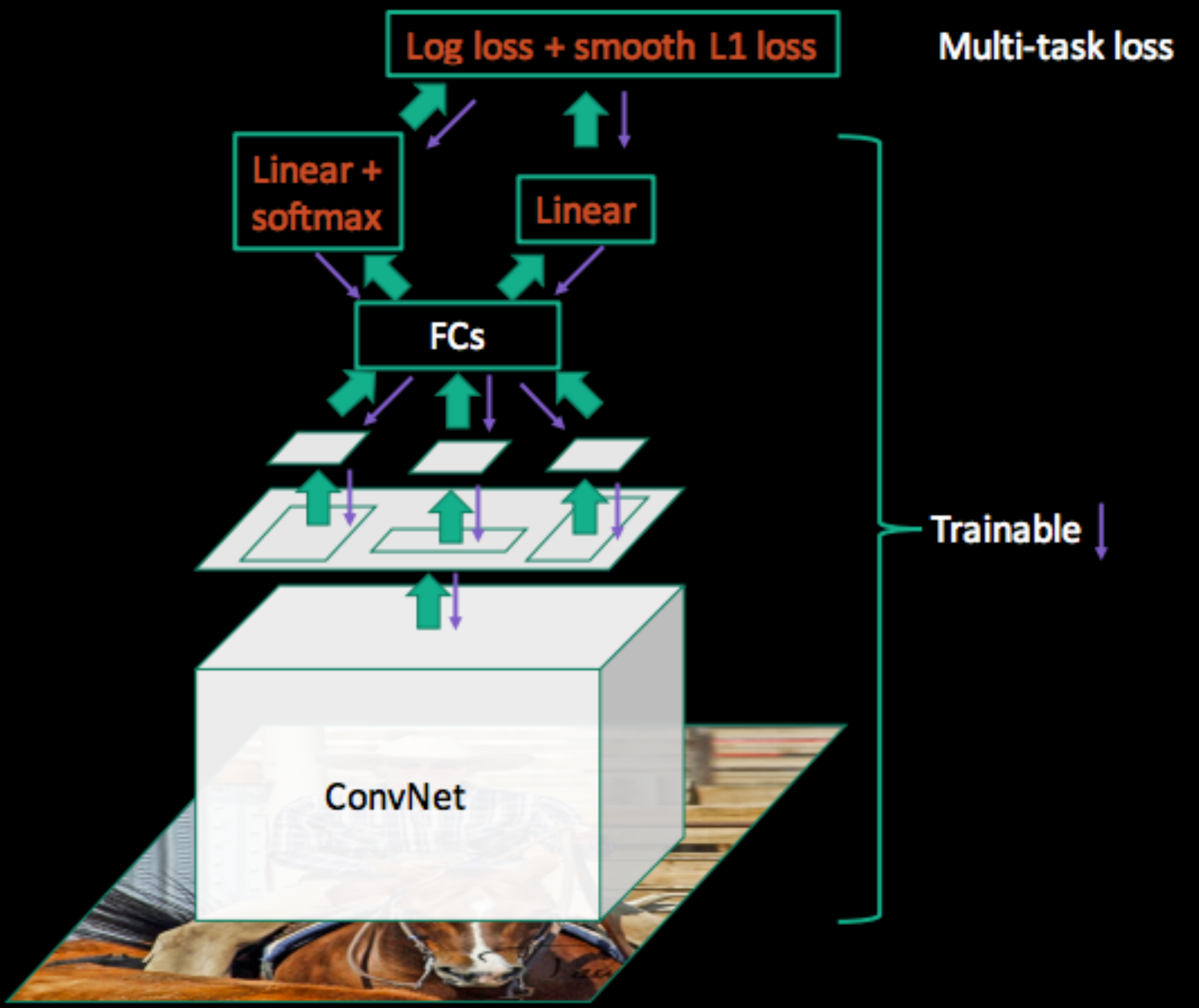
Object Detection System



Getting Proposals Feature Extraction **Classifier**

Fast R-CNN

Totally end-to-end!

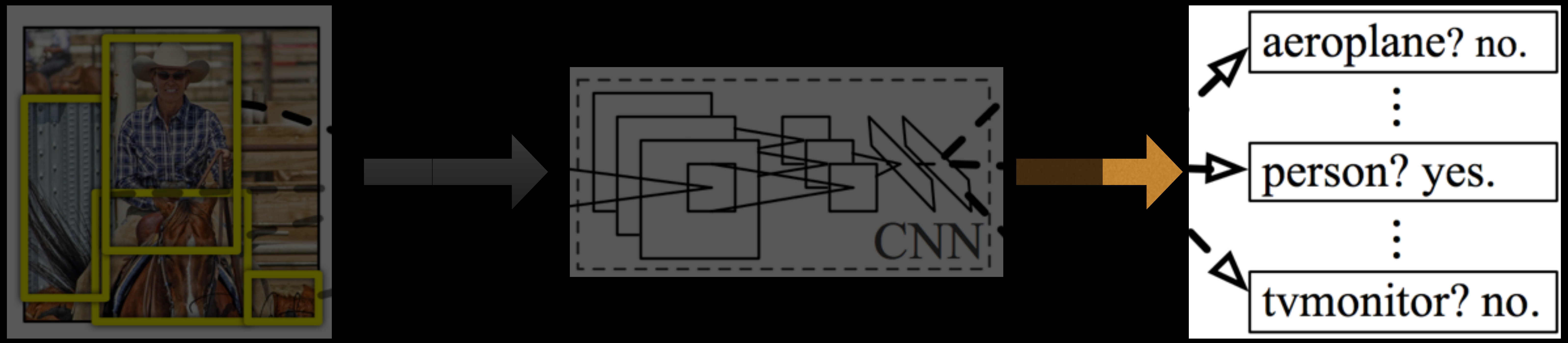


Slide credit: Ross Girshick

Fast R-CNN Results

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

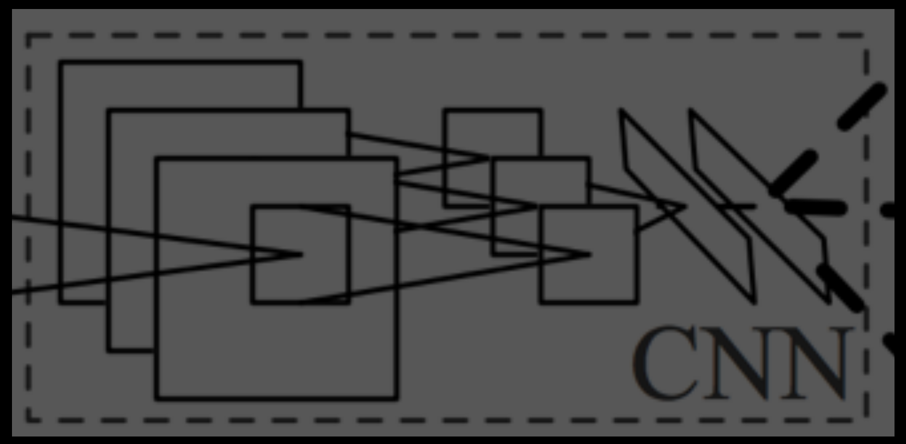
Object Detection System

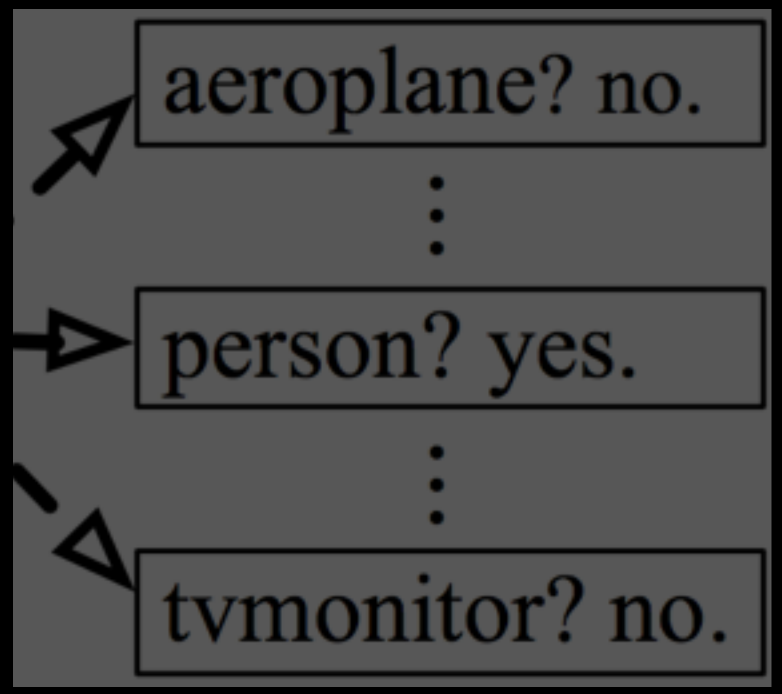


Getting Proposals Feature Extraction **Classifier**

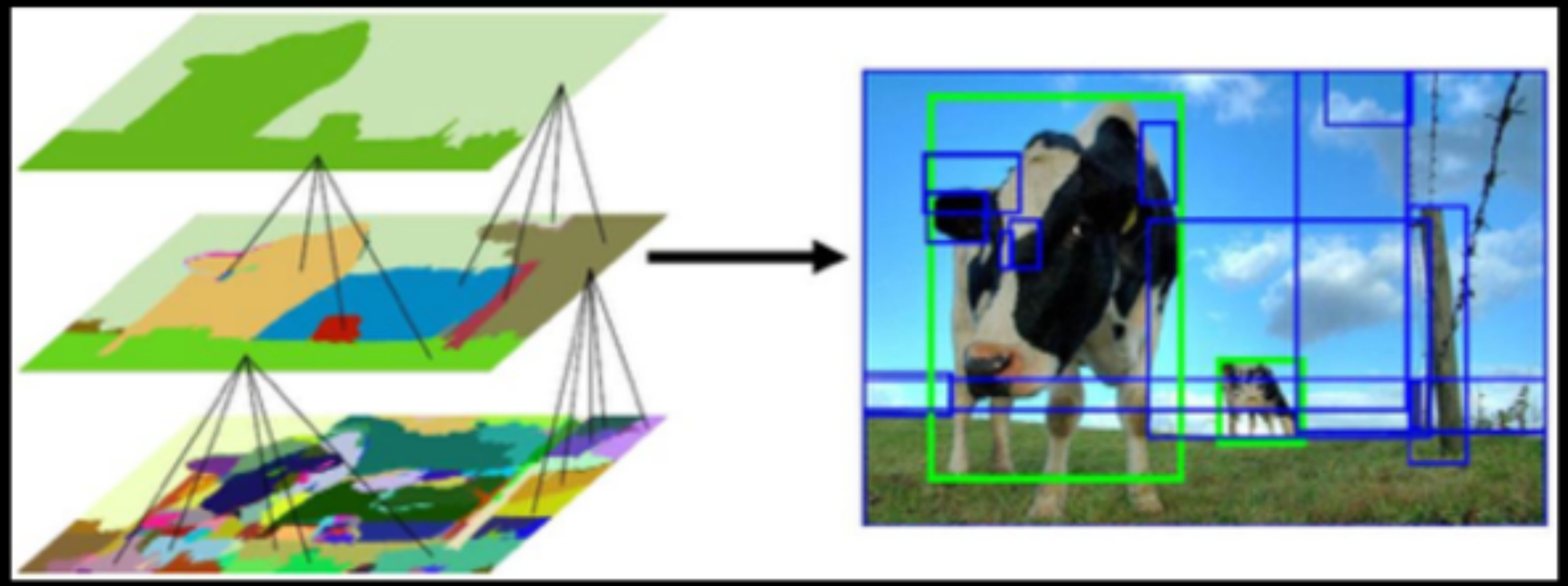
Fast R-CNN

Object Detection System



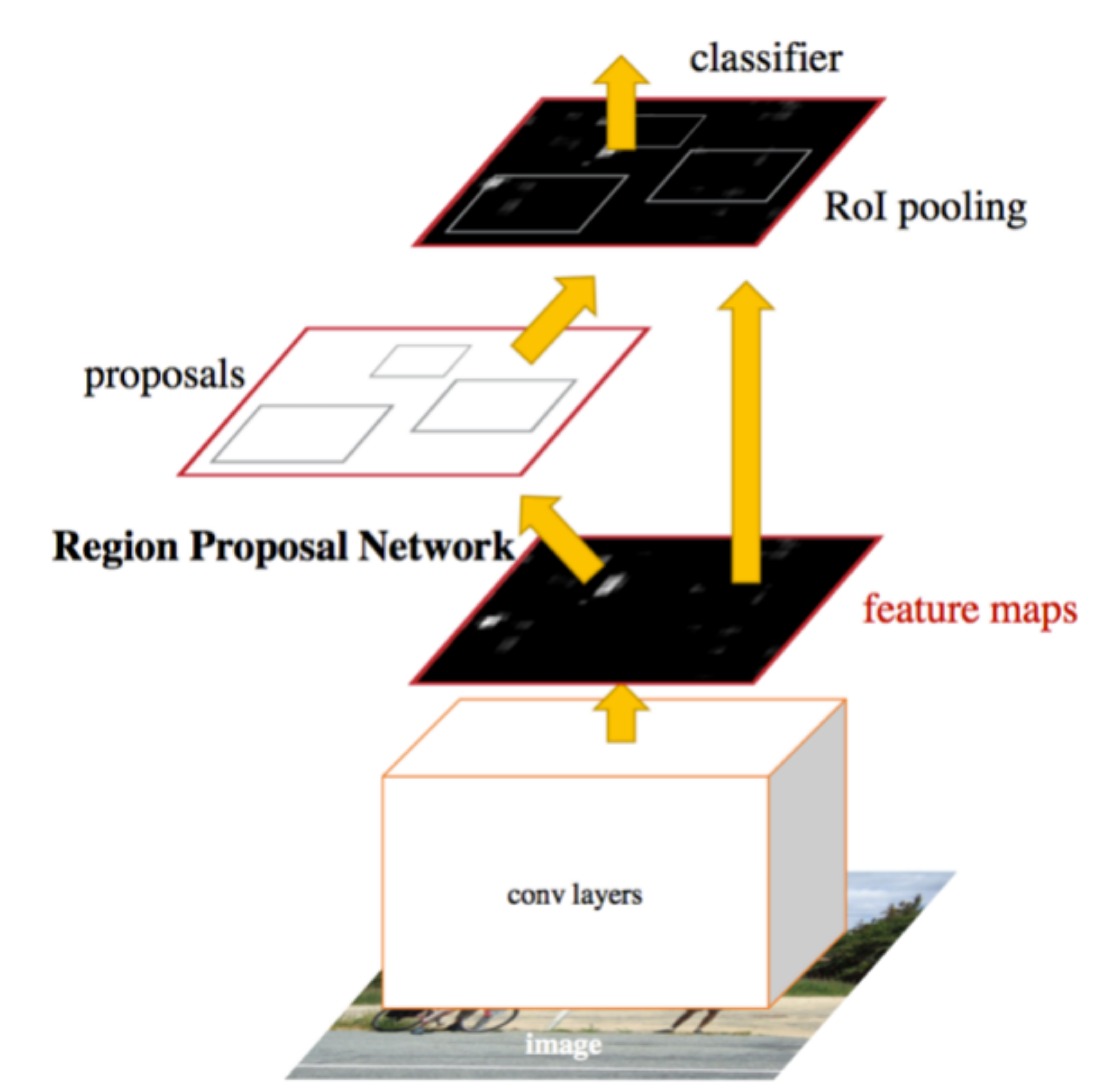


Getting Proposals Feature Extraction Classifier



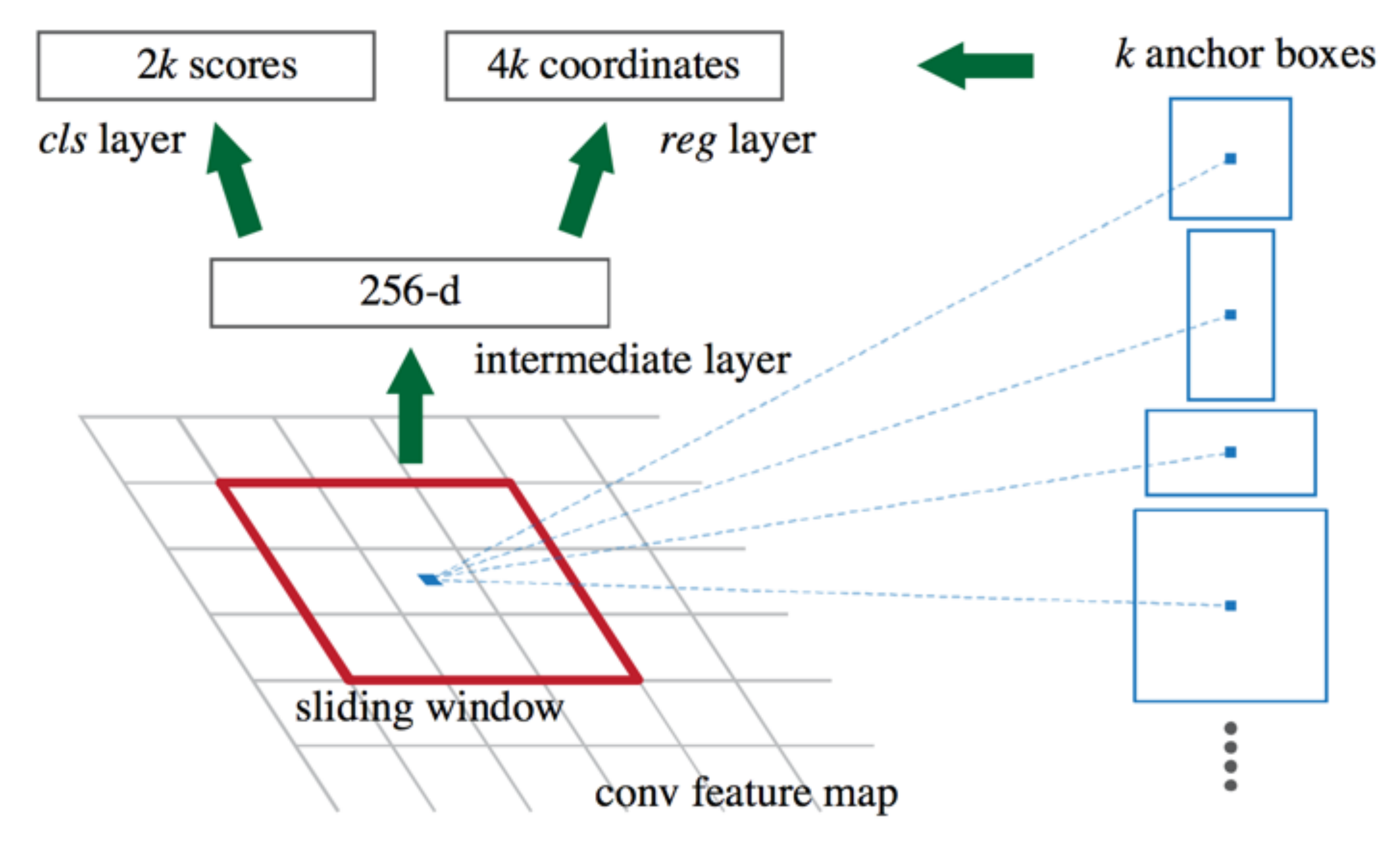
(e.g. selective search)

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

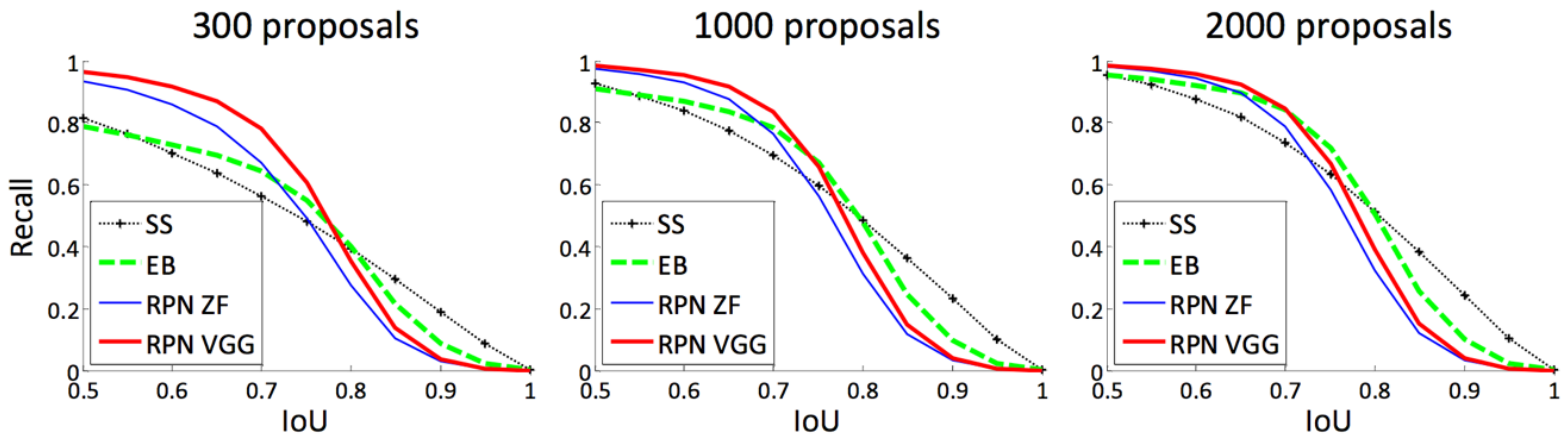


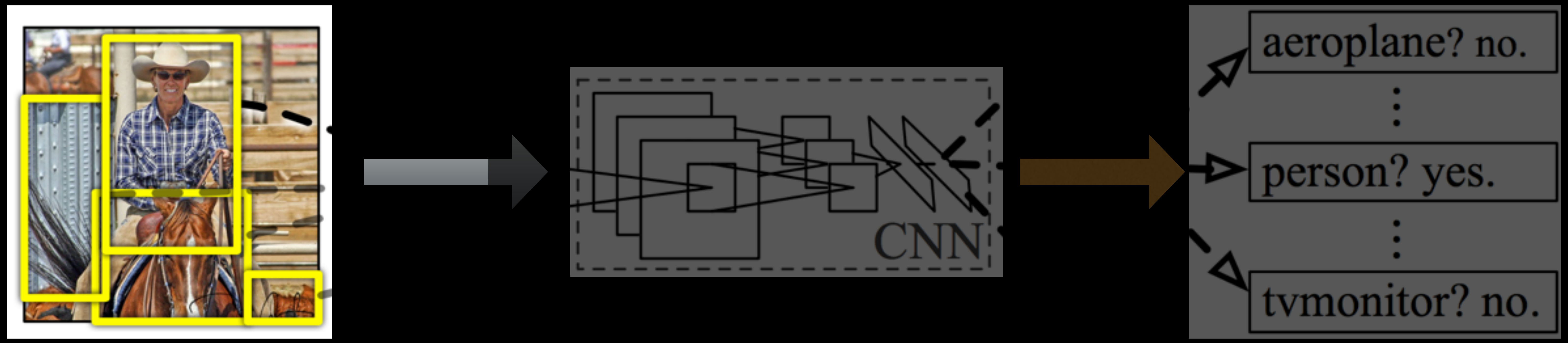
Figure 2: Recall *vs.* IoU overlap ratio on the PASCAL VOC 2007 test set.

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**

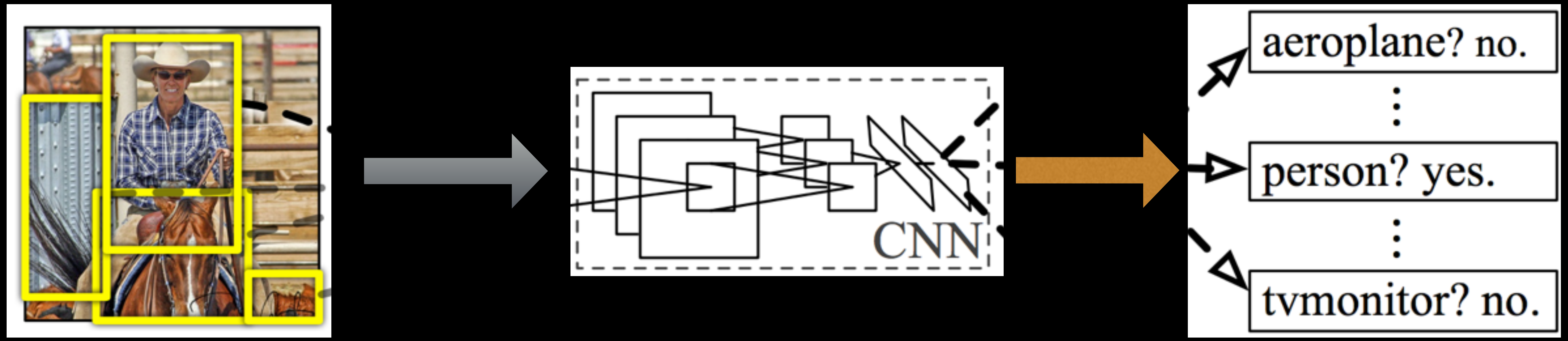
Object Detection System



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: Driving

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

⅓ Mile, 1760 feet

Slide credit: Joseph Chet Redmon

Example: Driving

	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

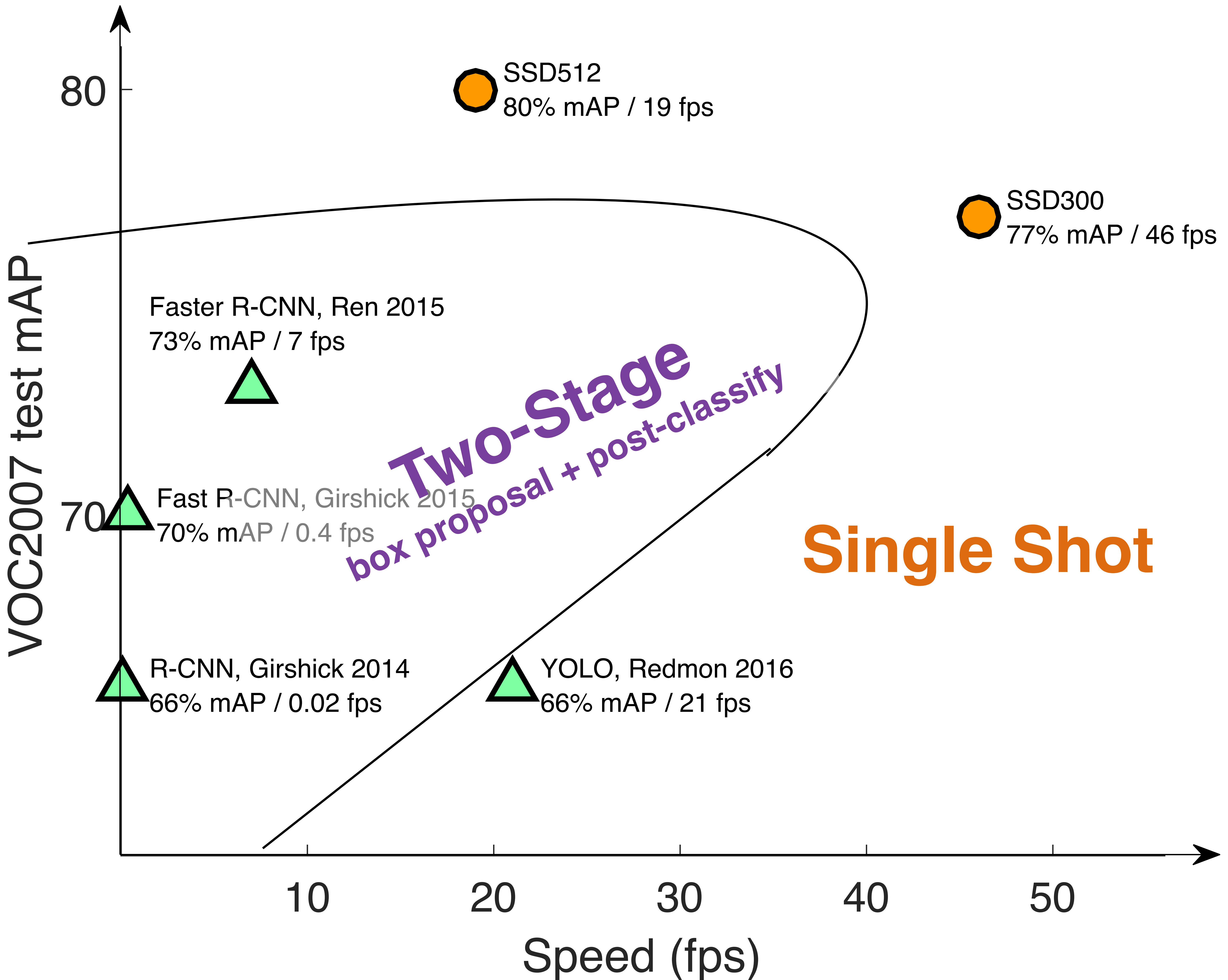
Slide credit: Joseph Chet Redmon

Example: Driving

	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

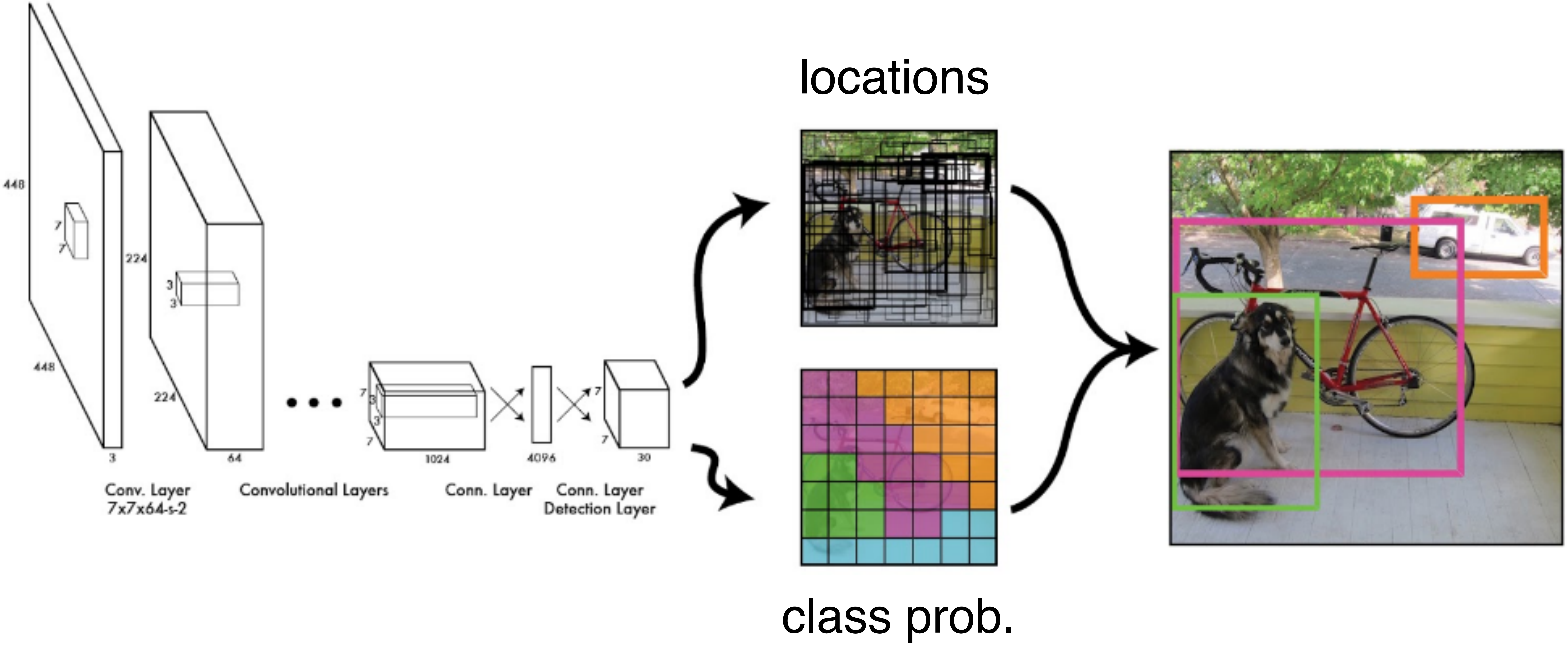
Slide credit: Joseph Chet Redmon

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

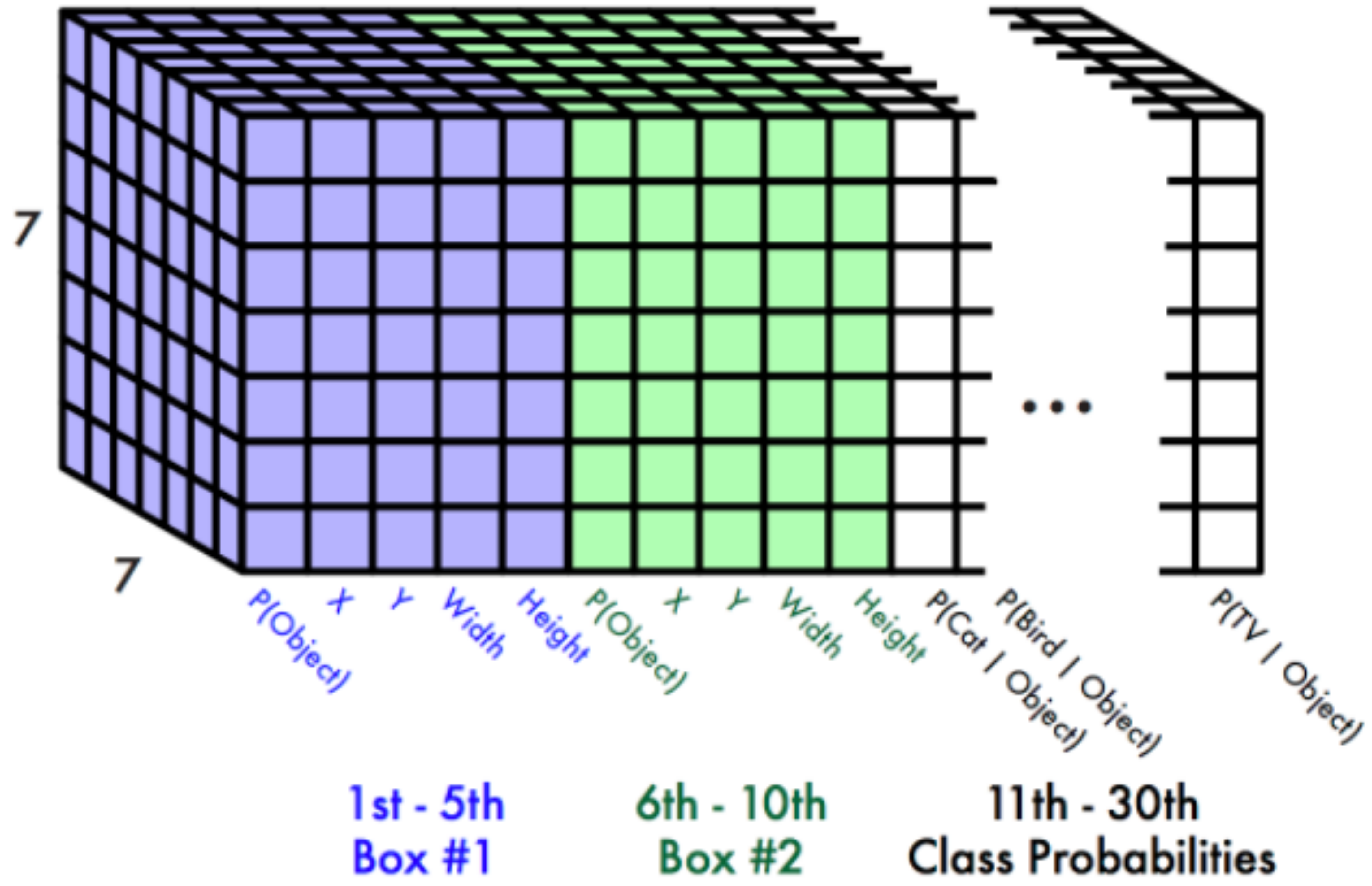
YOLO: output parameterization

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

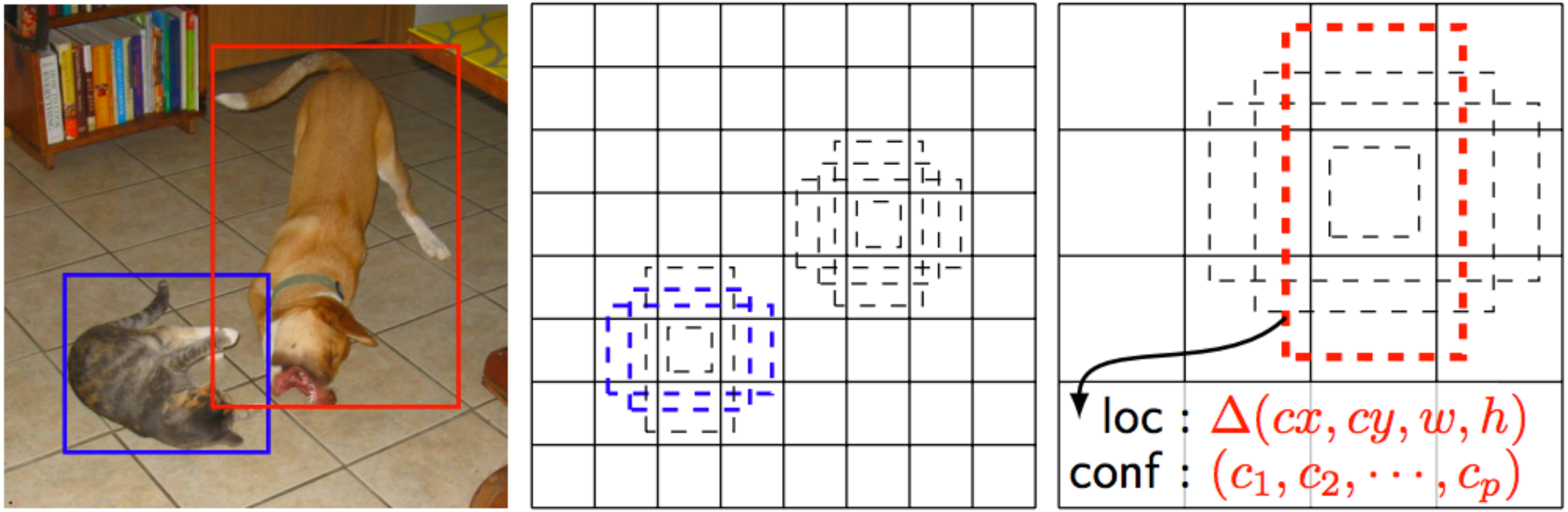
SSD: Single Shot MultiBox Detector

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

SSD: Single Shot MultiBox Detector

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

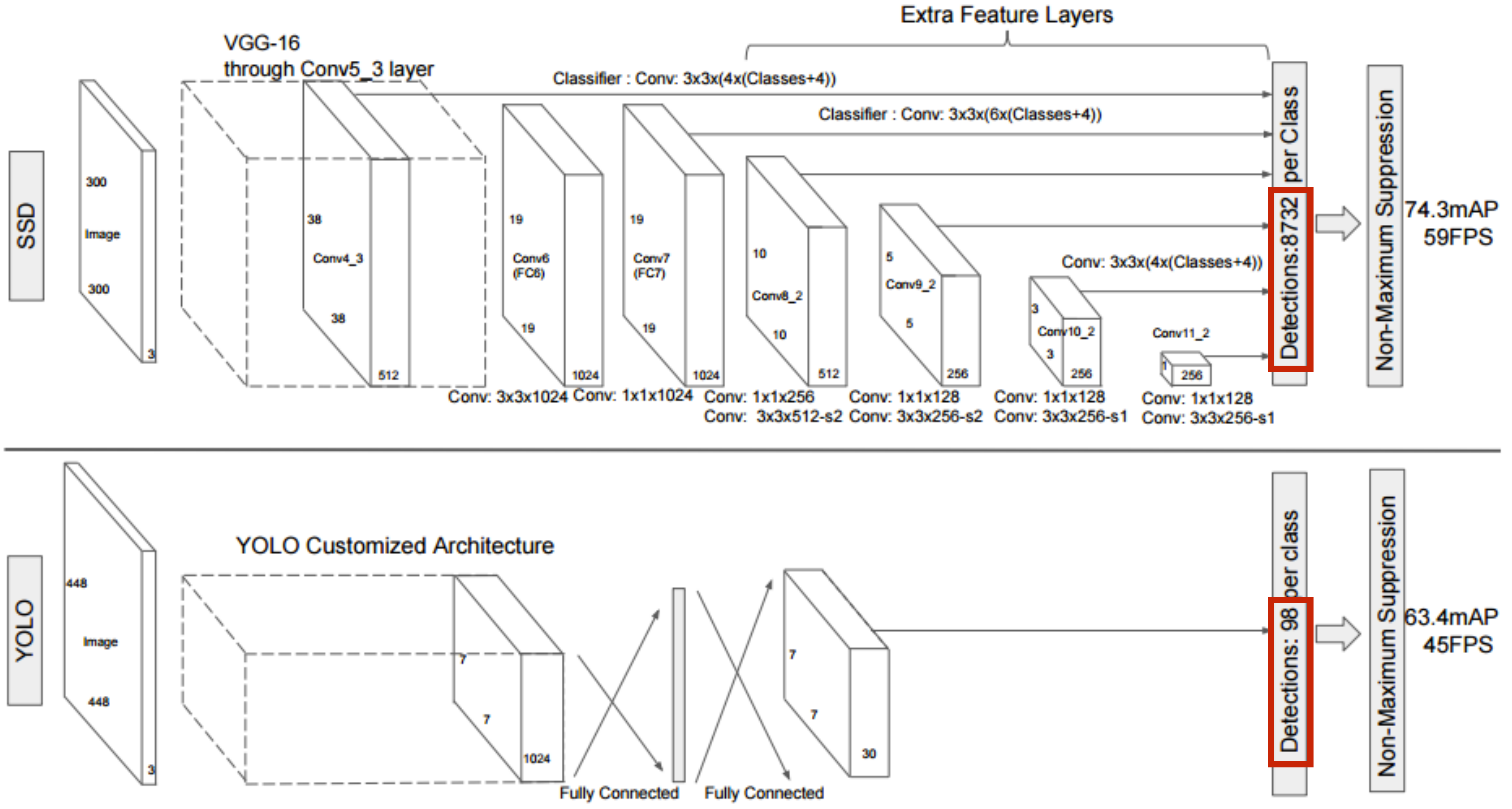
SSD: YOLO + default box shape + multi-scale



(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
- Face detection: from Viola-Jones to CNN
- General object detection
  - HOG detector
  - Deformable Part-based Model
  - Region-CNN
  - Fast versions of R-CNN
  - YOLO/SSD
- Future directions

Future directions:

Future directions:

- Real-time 3D object detection

Car

	Method	Setting	Code	Moderate	Easy	Hard	Runtime
1	AVOD			**85.44 %**	86.80 %	**77.73 %**	0.08 s
2	F-PointNet			84.00 %	88.70 %	75.33 %	0.17 s
3	DF-PC_CNN			80.69 %	88.89 %	76.04 %	0.5 s
4	NVLidarNet			80.04 %	84.44 %	74.31 %	0.1 s

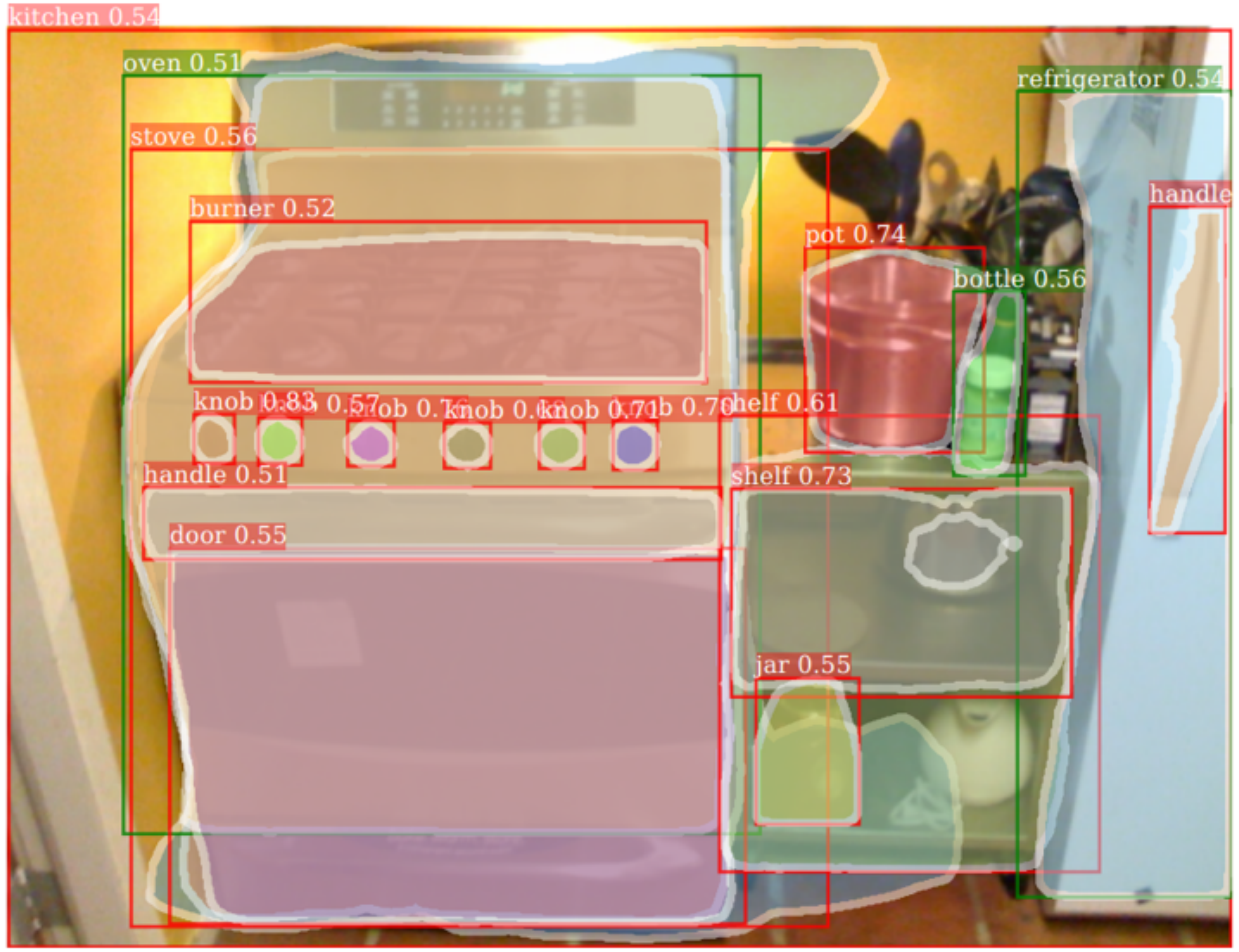
Future directions:

- Real-time 3D object detection
- Adversarial examples for object detection

Original Image Detected Whole Image Attacked Sign Region Attacked

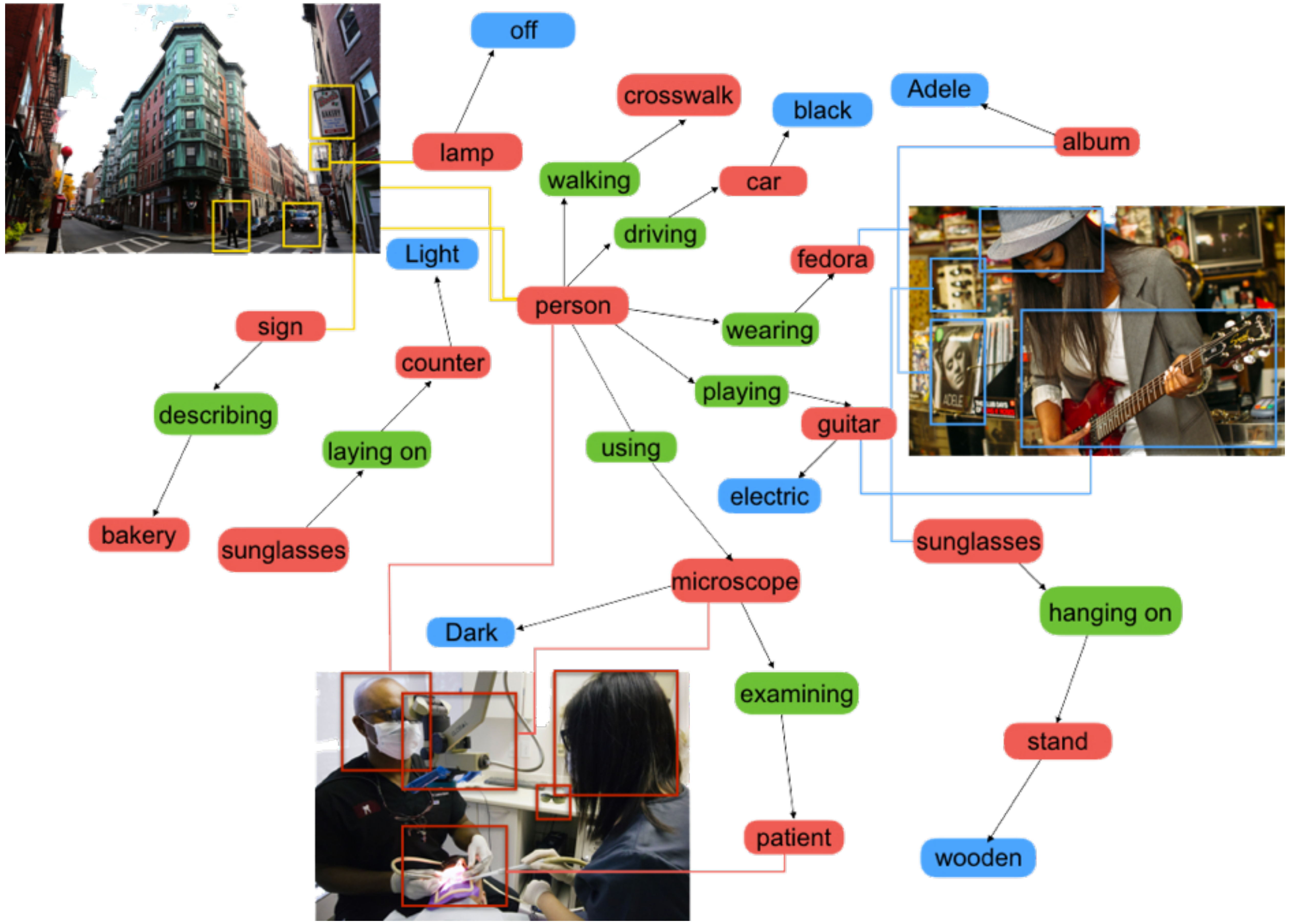
Future directions:

- Real-time 3D object detection
- Adversarial examples for object detection
- Weakly-supervised instance segmentation



Future directions:

- Real-time 3D object detection
- Adversarial examples for object detection
- Weakly-supervised instance segmentation
- High-level scene graph construction



Object detection

- Introduction
- Face detection: from Viola-Jones to CNN
- General object detection
  - HOG detector
  - Deformable Part-based Model
  - Region-CNN
  - Fast versions of R-CNN
  - YOLO/SSD
- Future directions