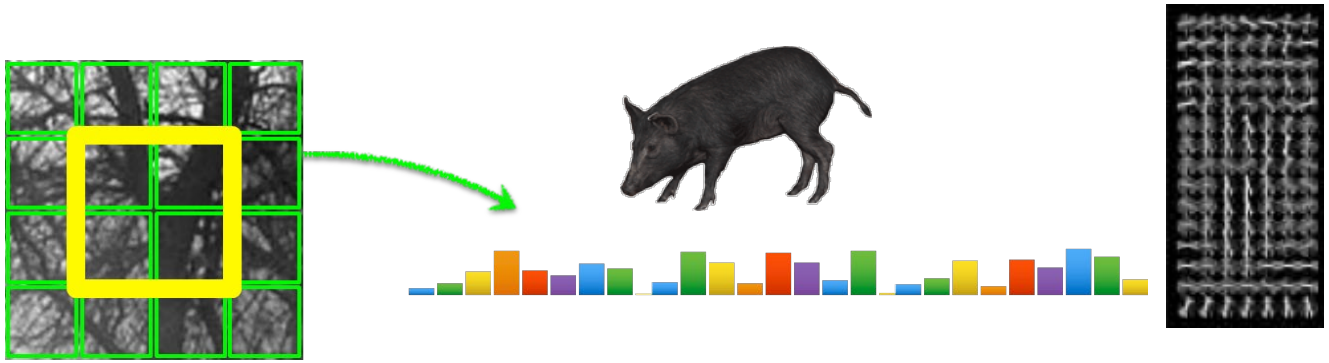
Introduction to Computer Vision

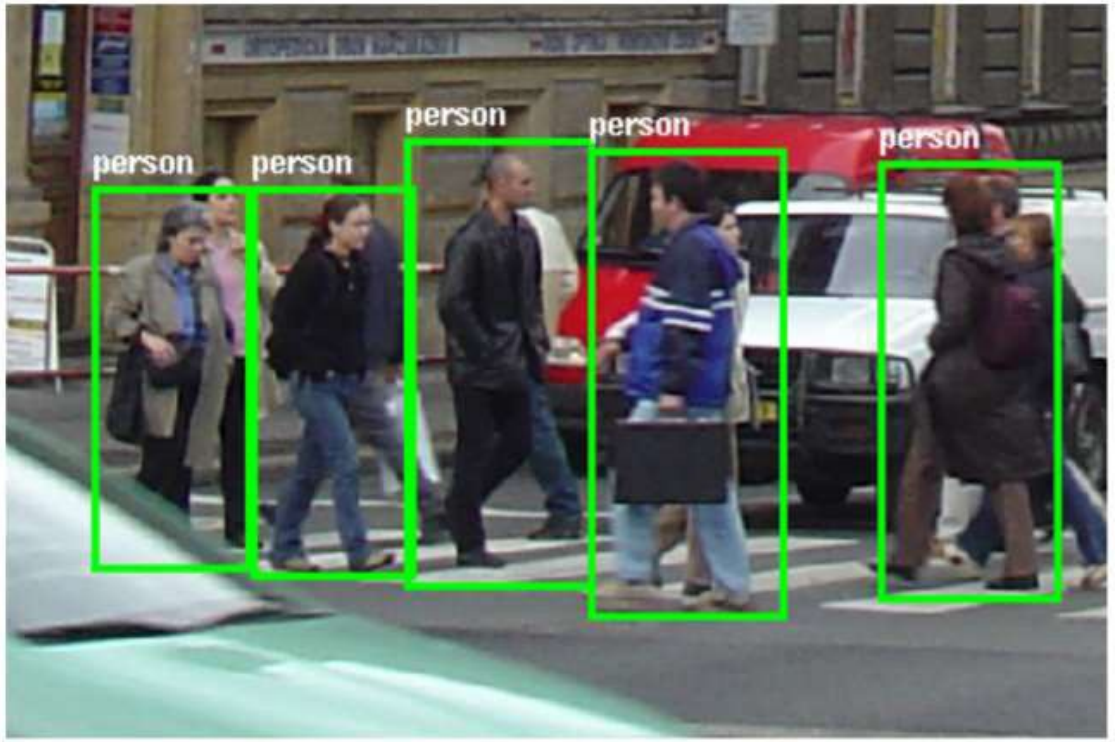
Kaveh Fathian
Assistant Professor
Computer Science Department
Colorado School of Mines

Lecture 16

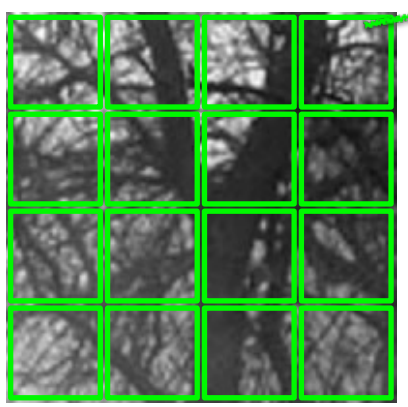
Learning Objectives

- HOG Descriptors
- HOG Pedestrian Detection





HOG Descriptors



8x8 cell

30	20	10	…				
40	60	70					
50	70	60					
⋮			⋱				

	20		…				
40	60	70					
	70						
⋮			⋱				

Compute gradient:

Outperforms

centered

-1	0	1

uncentered

-1	1

diagonal

0	1
-1	0

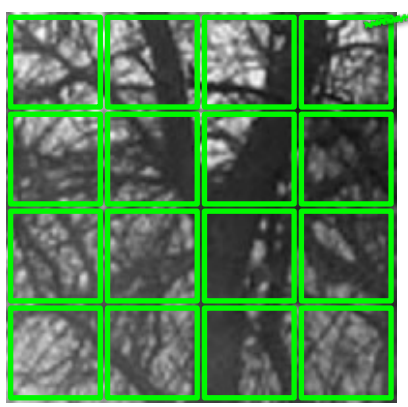
cubic-corrected

1	-8	0	8	-1

Sobel

-1	0	1
-2	0	2
-1	0	1

HOG Descriptors



8x8 cell

30	20	10	…				
40	60	70					
50	70	60					
⋮			⋱				

	20		…				
40	60	70					
	70						
⋮			⋱				

Compute gradient

Outperforms

-1	0	1

centered

X dir: $-40 + 70 = 30$

Y dir: $-20 + 70 = 50$

Grad mag: $\sqrt{30^2 + 50^2} = 58$

Grad deg: $\tan^{-1}(50/30) = 60^o$

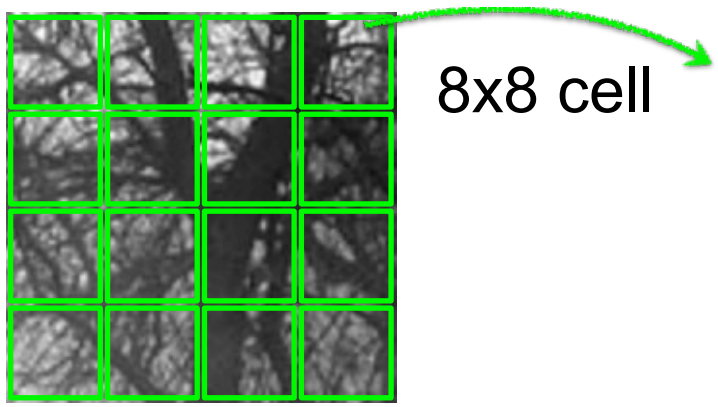
			…				
	58						
⋮			⋱				

Grad magnitude

			…				
	60						
⋮			⋱				

Grad angle

HOG Descriptors



8x8 cell

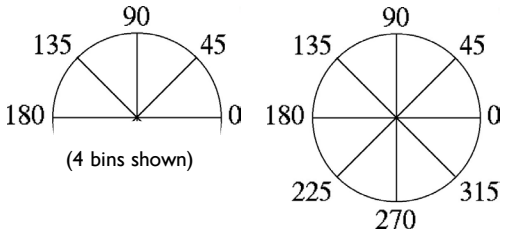
	58		…				
⋮			⋱				
	60		60		60		

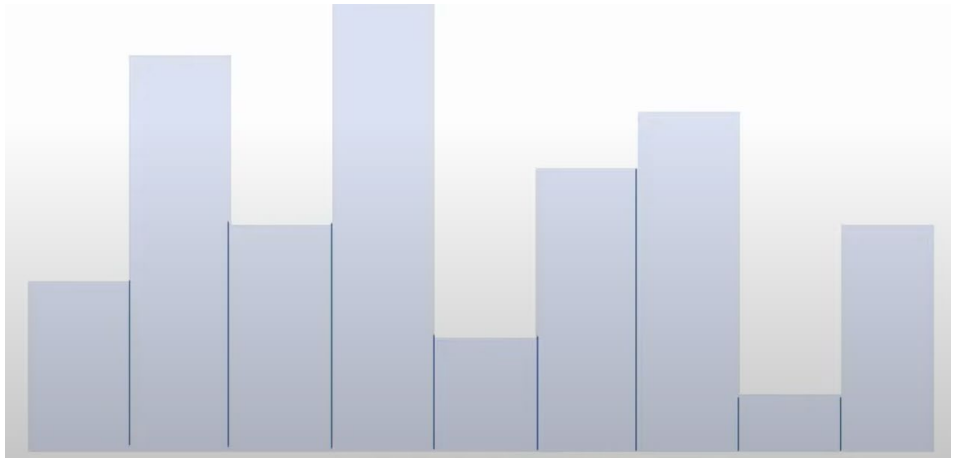
Grad magnitude

	60		…				
⋮			⋱				
	20		40		30		

Grad angle

Orientation: 9 bins
(for unsigned angles 0-180)

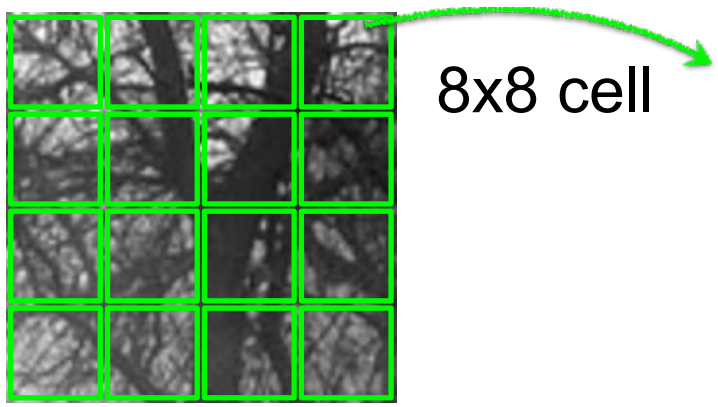




Outperforms

Unsigned angles:	0	20	40	60	80	100	120	140	160
Signed angles:	0	40	80	160	200	240	280	320	360

HOG Descriptors

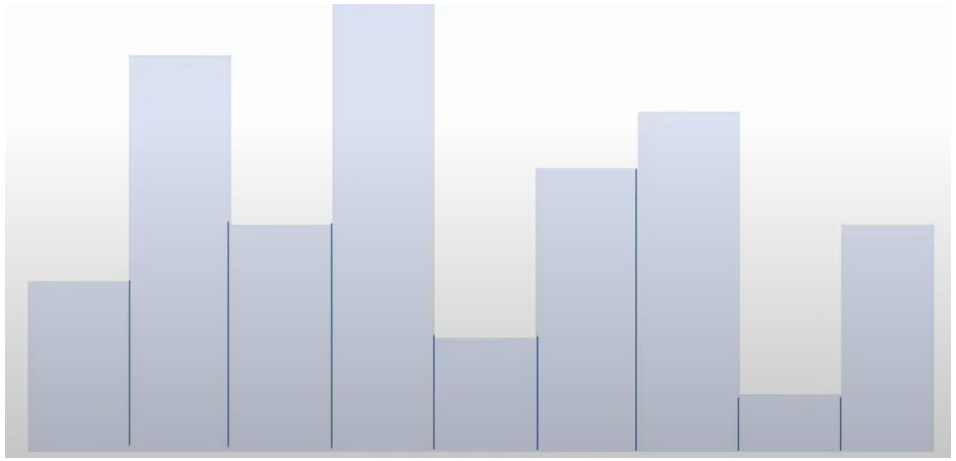


			…				
	58						
⋮			⋱				
	60		60		60		

Grad magnitude

			…				
	60						
⋮			⋱				
	20		40		30		

Grad angle

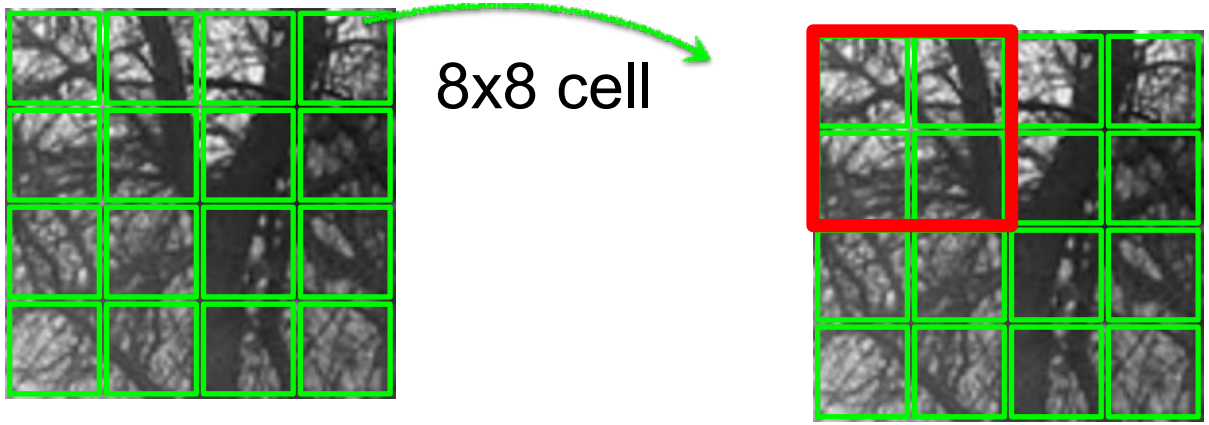


Unsigned angles:	0	20	40	60	80	100	120	140	160
Signed angles:	0	40	80	160	200	240	280	320	360

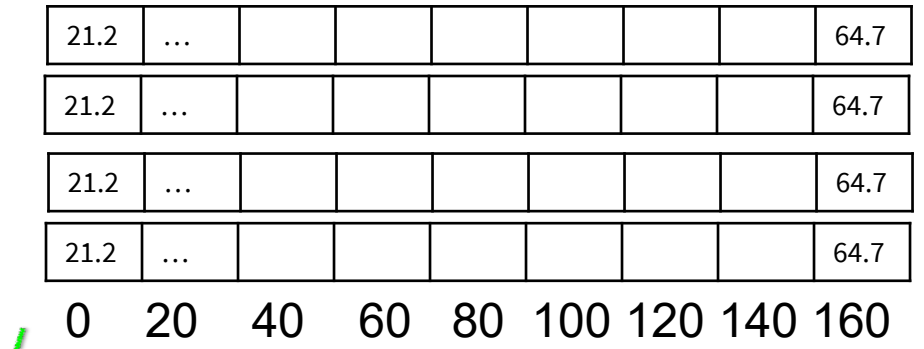
Descriptor for cell

21.2	…							64.7
0	20	40	60	80	100	120	140	160

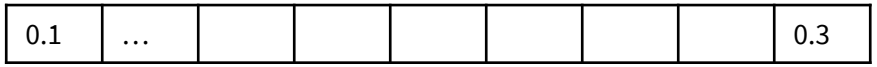
HOG Descriptors



Group 2x2 cells into blocks and normalize their descriptor:



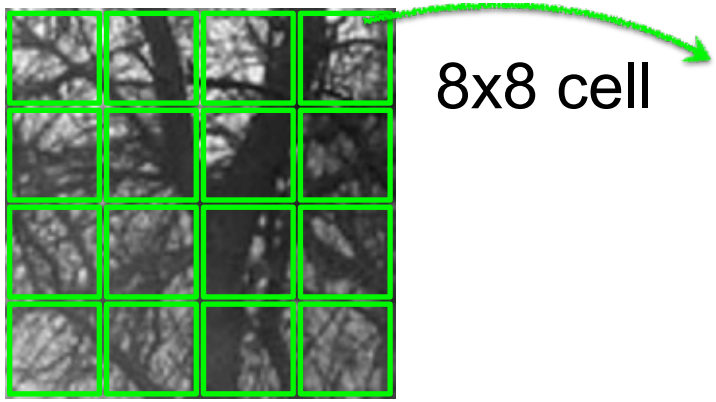
Group & normalize (L2 norm)



HOG descriptor

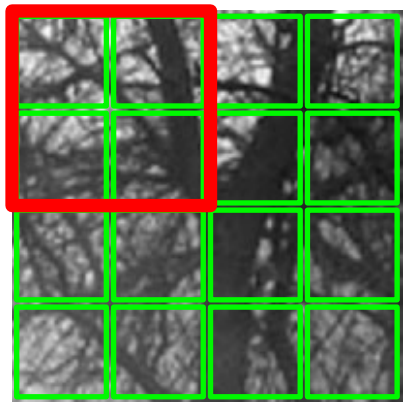
Each block produces a vector:
(4 cells) × (9 bins) = 36-dimensional block descriptor

HOG Descriptors

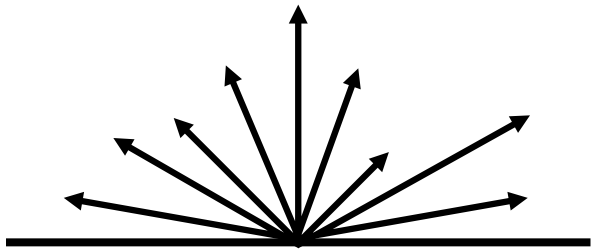


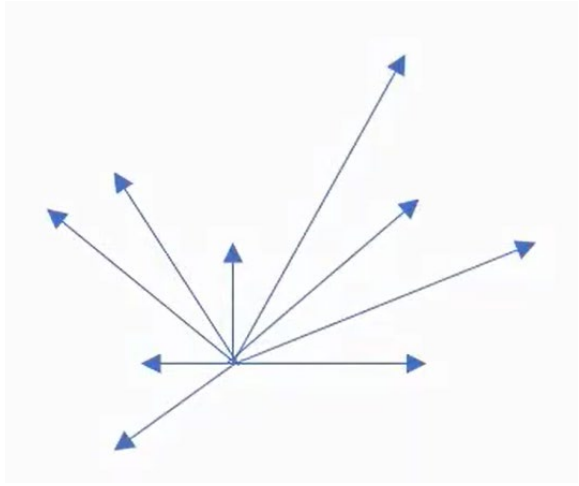
8x8 cell

block



HOG descriptor
for block



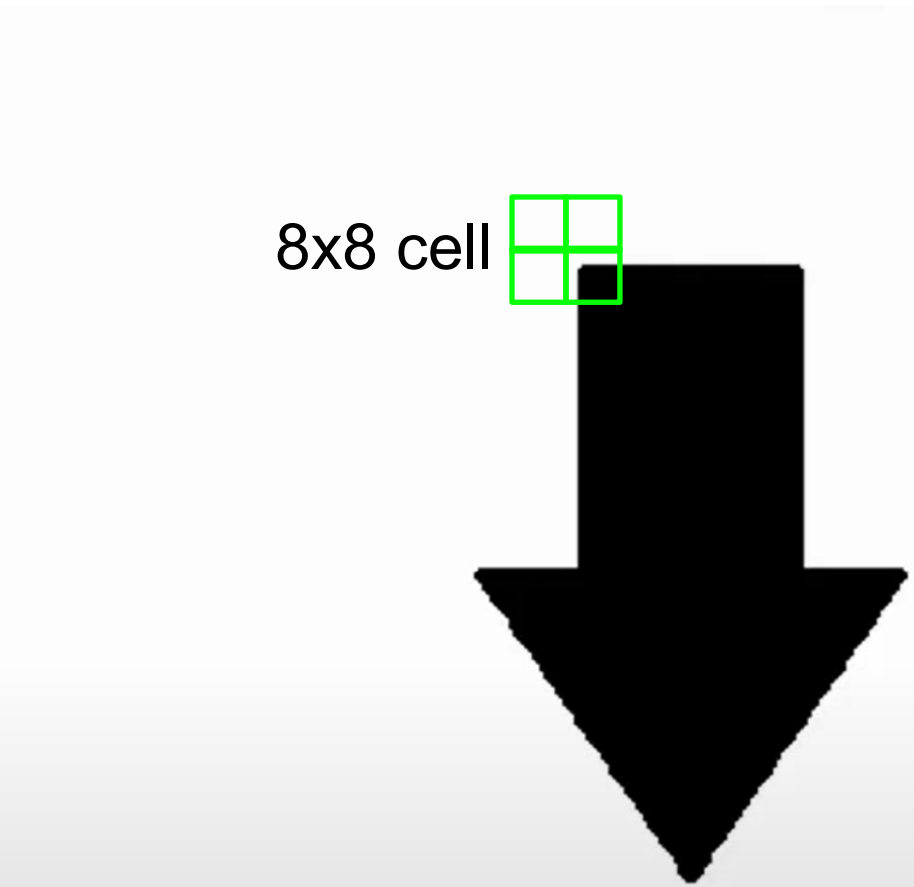


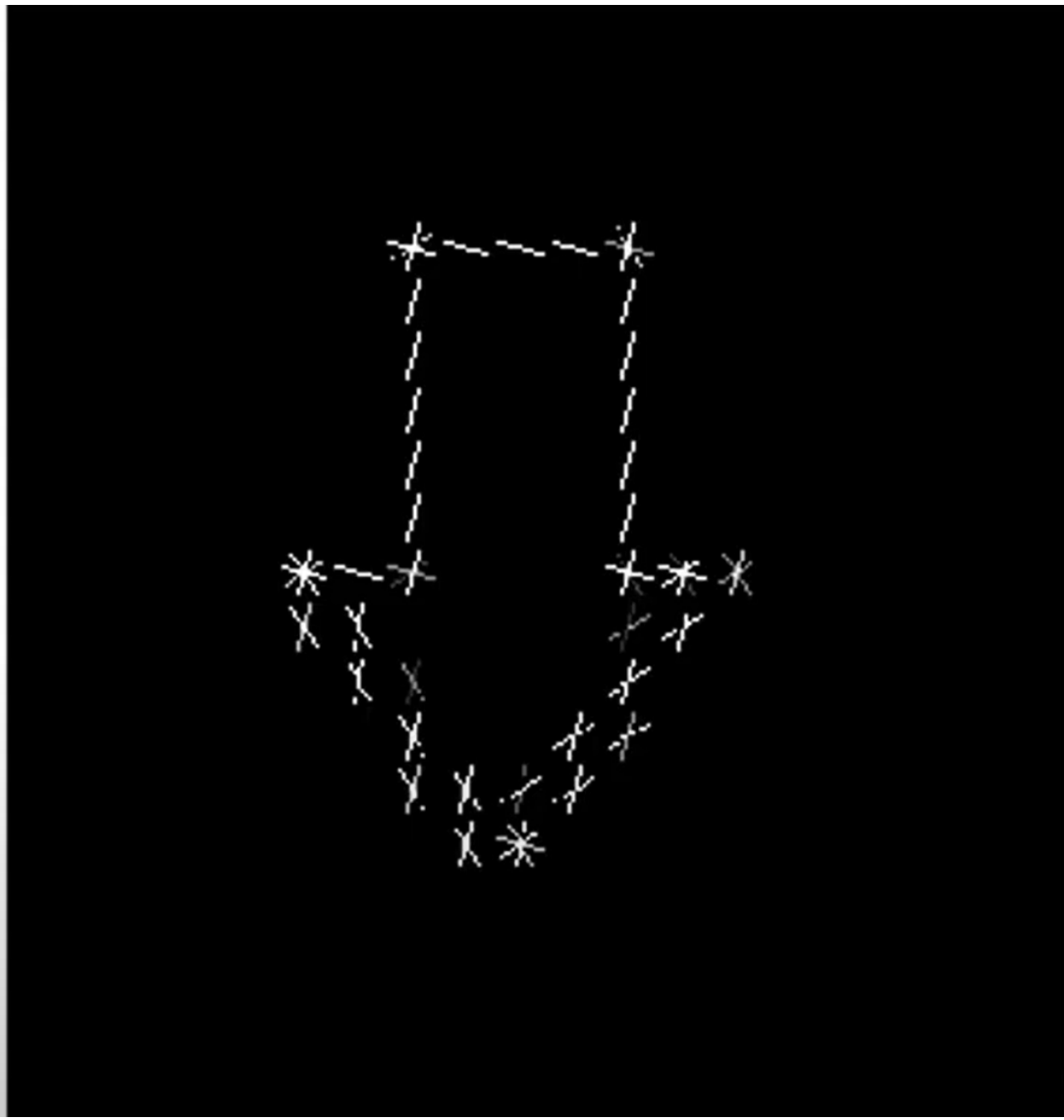
Visualization

HOG Example

Input Image

HOG Descriptors



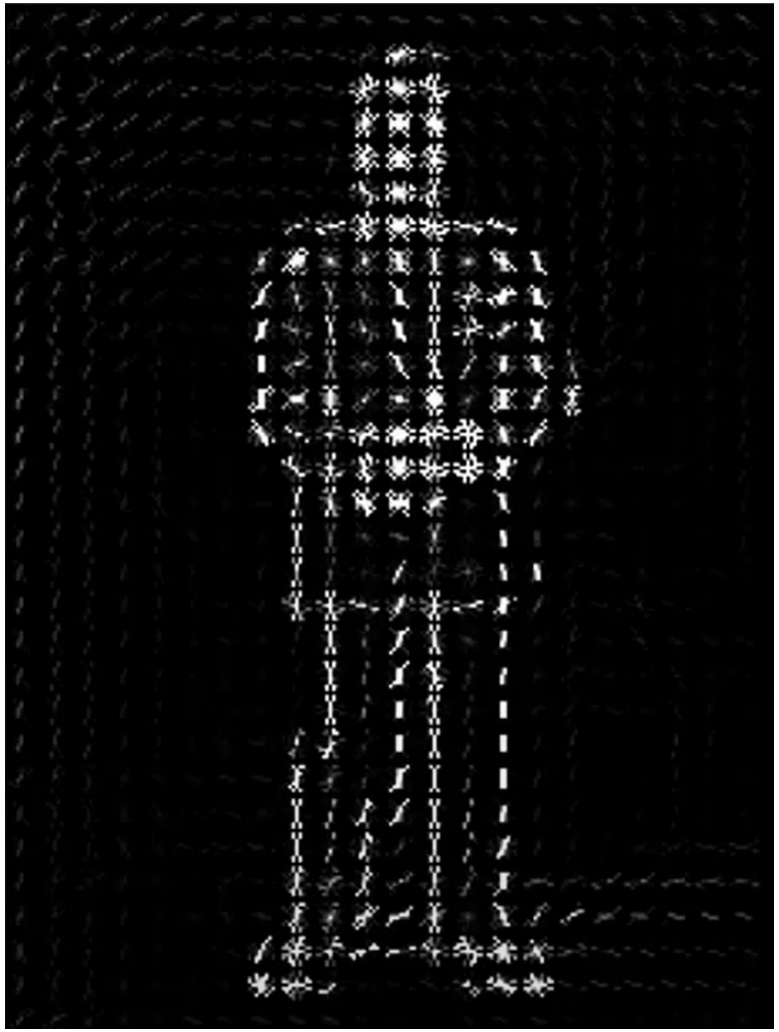


HOG Example

Input Image

8x8 cell

HOG Descriptors



Dalal-Triggs Pedestrian Detector

Algorithm

1. Extract fixed-sized (64x128 pixel) window at each position and scale
2. Compute HOG (histogram of oriented gradient) features within each window
3. Score the window with a linear SVM classifier
4. Perform non-maxima suppression to remove overlapping detections with lower scores

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

Extracting HOG Features

150x300

64x128

8x16

8x8 cell

30	20	10	…				
40	60	70					
50	70	60					
⋮			⋱				

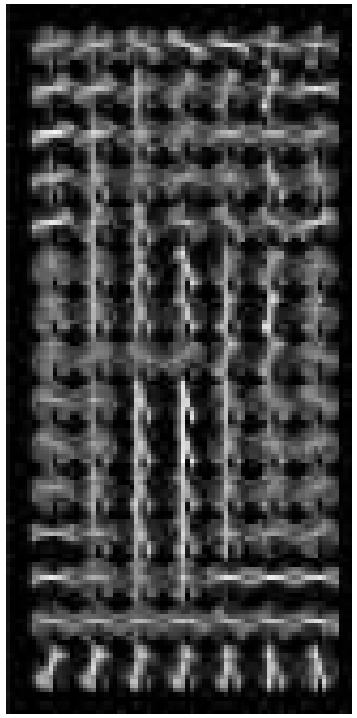
HOG Pedestrian Detection

1 cell step size

128 pixels
16 cells
15 blocks

64 pixels
8 cells
7 blocks

visualization

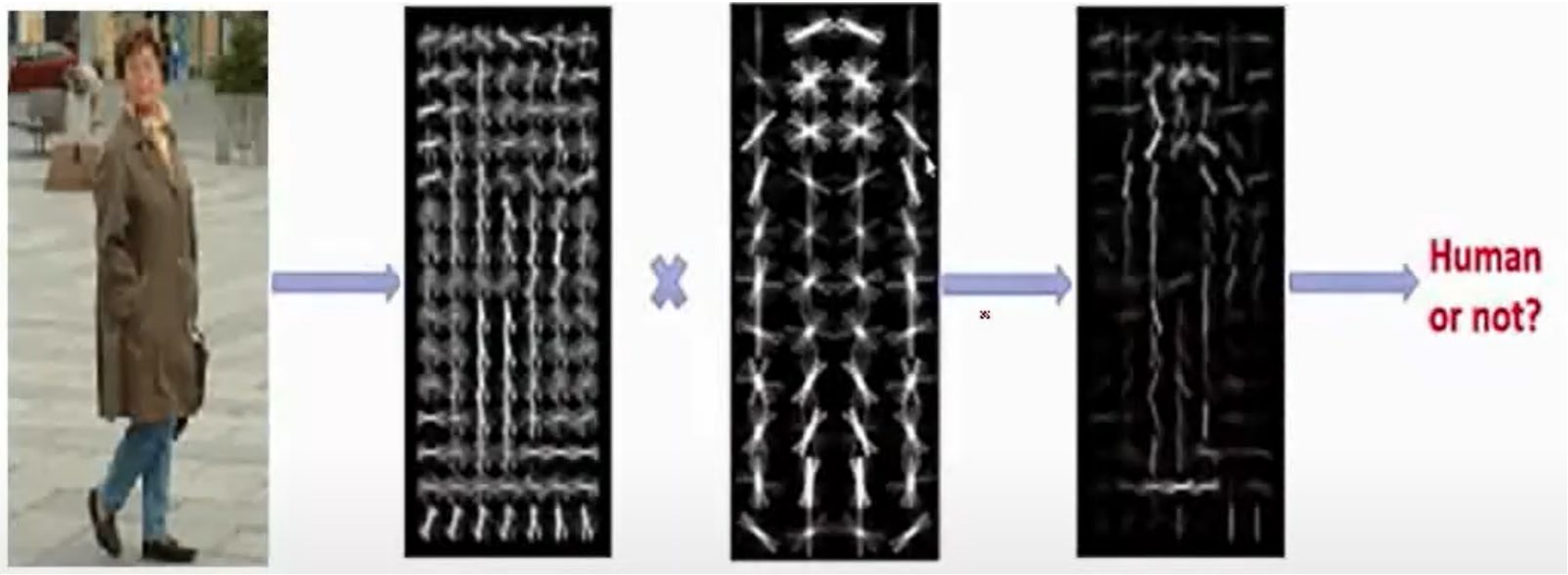


15 x 7 x 4 x 9 =
3780

Redundant representation due to overlapping blocks

http://chrisjmccormick.wordpress.com/2013/05/09/hog-person-detector-tutorial/

HOG Pedestrian Detection

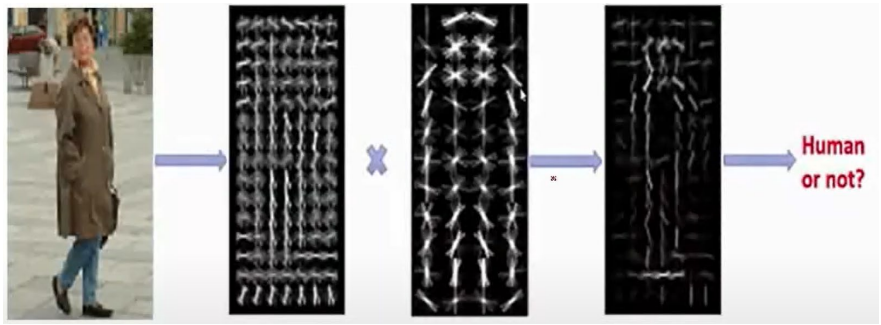


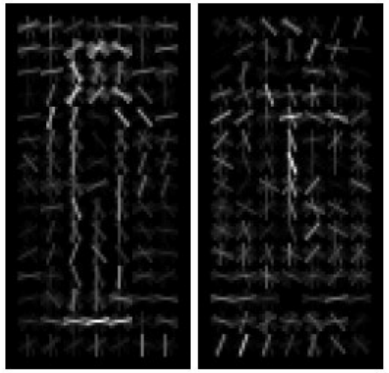
Image

HOG Features

Human Model

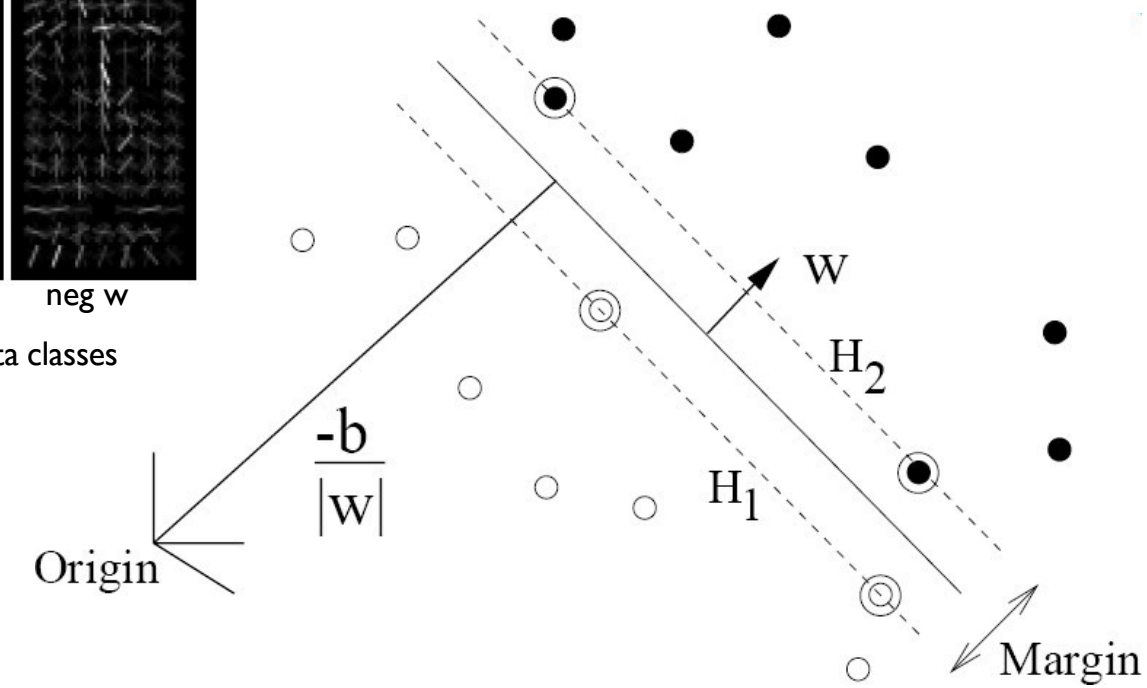
Output of convolving with model

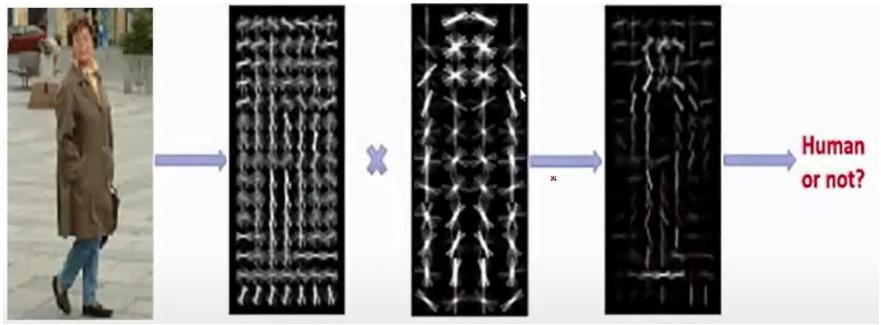


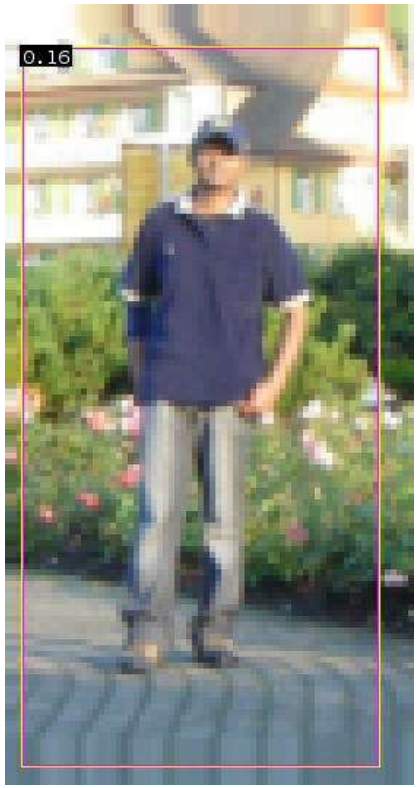


pos w neg w

Training data classes







$$0.16 = w^T x - b$$

$$sign(0.16) = 1$$

$=>$ pedestrian

Multi-Scale Detection