

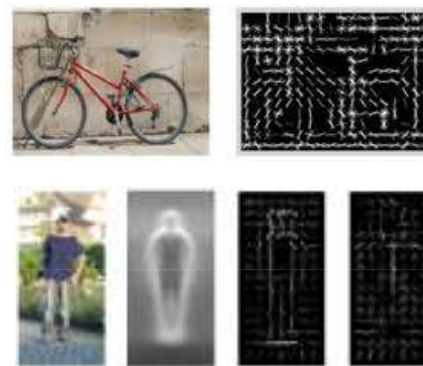
Histogram of Oriented Gradients

261458 & 261753 Computer Vision

#4

HOG

Histogram of Oriented Gradients



N. Dalal & B. Triggs, CVPR 2005

- One of the state-of-the-art shape-based image feature
- Used for pedestrian detection
- Use for film & media analysis
- Used for visual surveillance
- Fast calculation
- Less detailed so that it consumes small amount of memory and hard-disk

261458 & 261753 Computer Vision

#4

Gradient

3

Original Image



STEP 1: Calculate image gradients

261458 & 261753 Computer Vision

#4

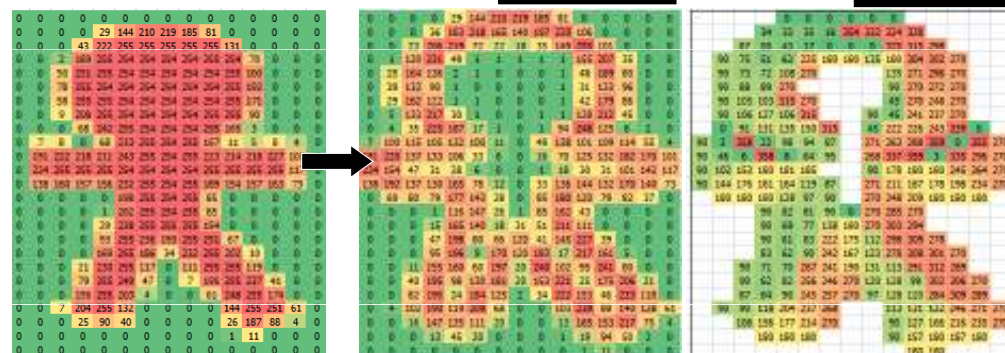
Gradient

4

Magnitude

(360° Orientation System)

Direction



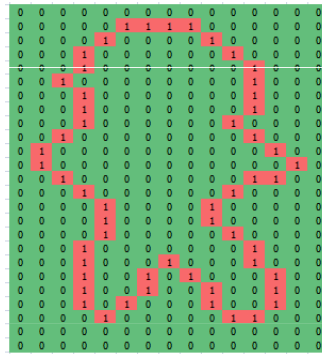
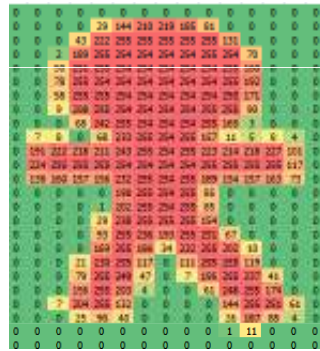
STEP 1: Calculate image gradients

261458 & 261753 Computer Vision

#4

Edge

5

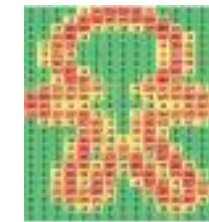
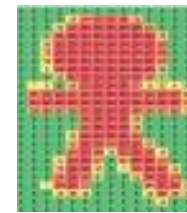


Edge Image

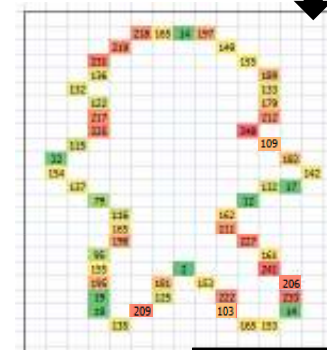
STEP 2: Edge detection

261458 & 261753 Computer Vision

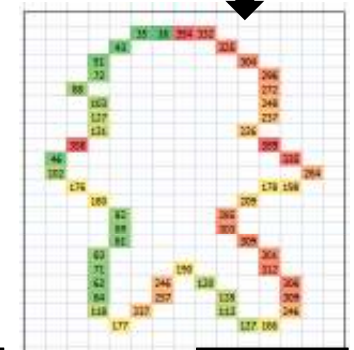
#4



STEP 3:
Masking
gradient
with edge



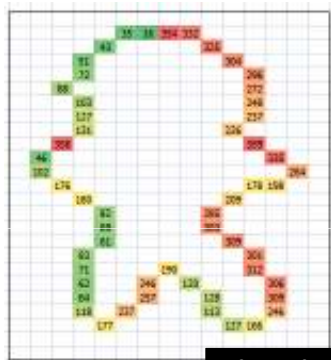
Magnitude



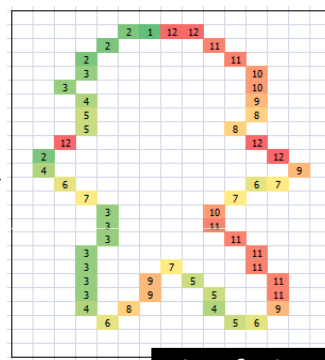
Direction

261458 & 261753 Computer Vision

#4



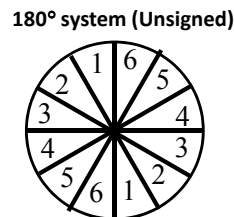
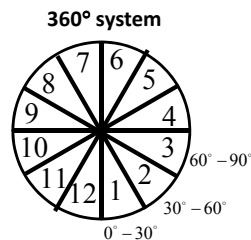
Direction



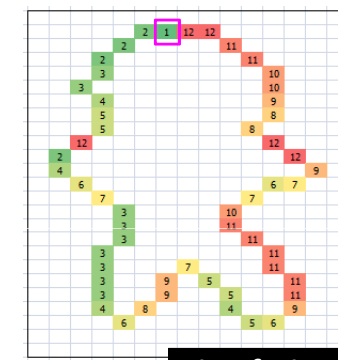
Bin of Direction

(This
example
use 12 bins
in 360°
orientation
system)

**STEP 4: Quantize
gradient
direction
into several bins**



Magnitude



Bin of Direction

**STEP 5: Collect the gradient
magnitude for each bins**

165

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

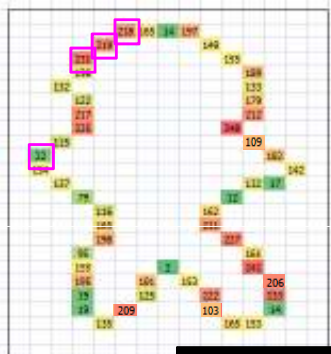
12 bins, 360° system

261458 & 261753 Computer Vision

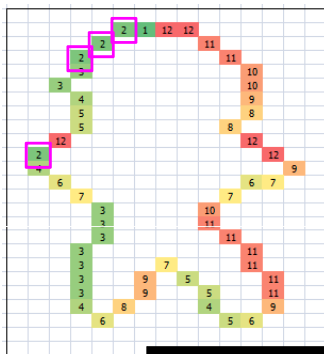
#4

261458 & 261753 Computer Vision

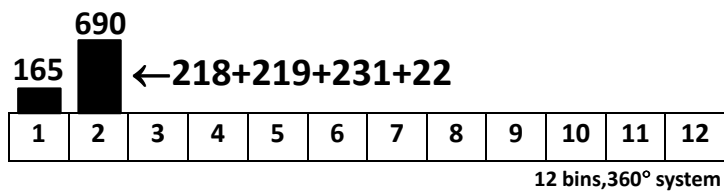
#4



Magnitude



Bin of Direction

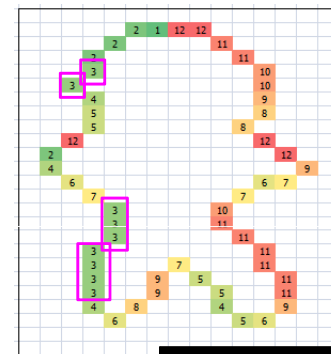


261458 & 261753 Computer Vision

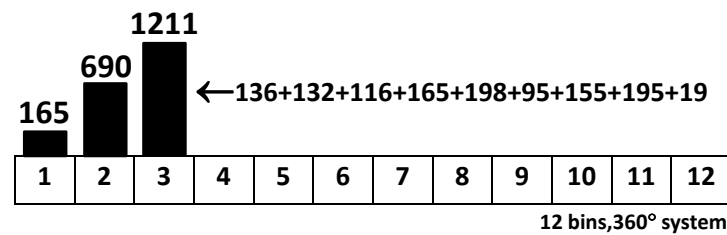
#4



Magnitude

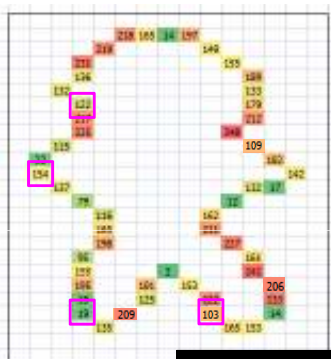


Bin of Direction

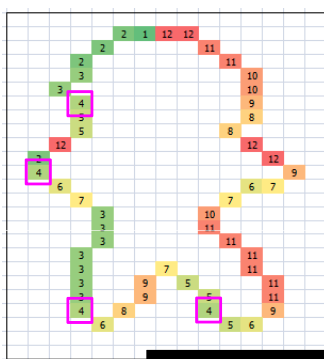


261458 & 261753 Computer Vision

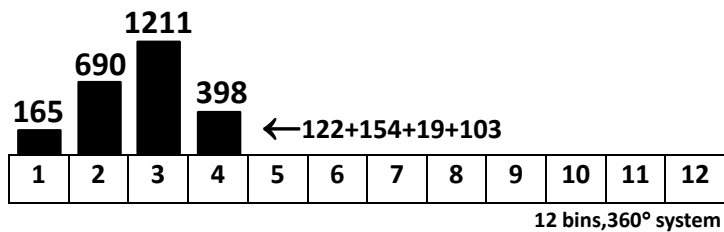
#4



Magnitude

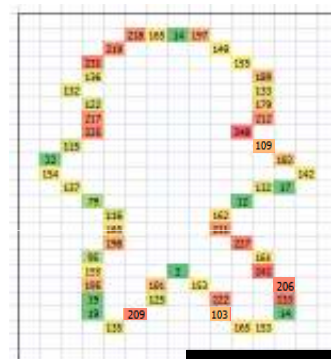


Bin of Direction

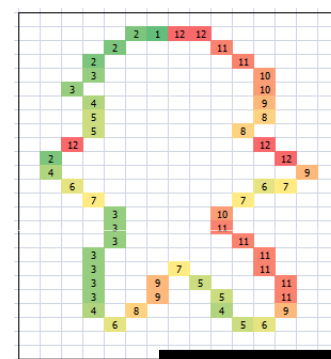


261458 & 261753 Computer Vision

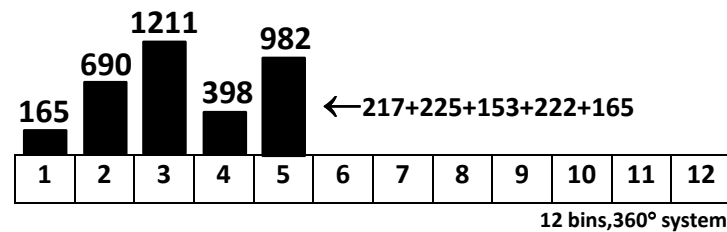
#4



Magnitude

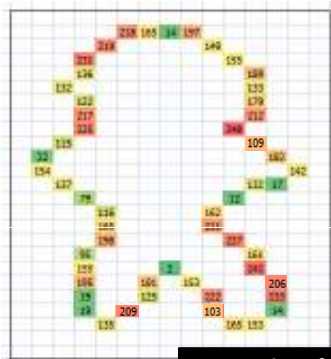


Bin of Direction

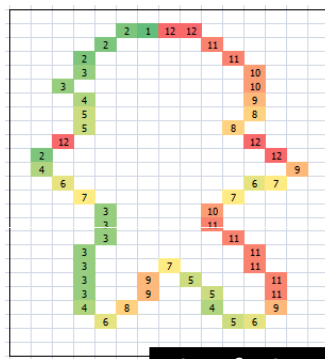


261458 & 261753 Computer Vision

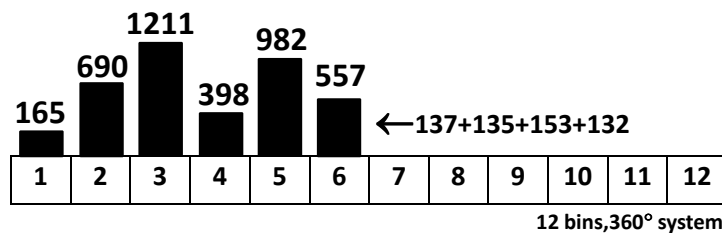
#4



Magnitude

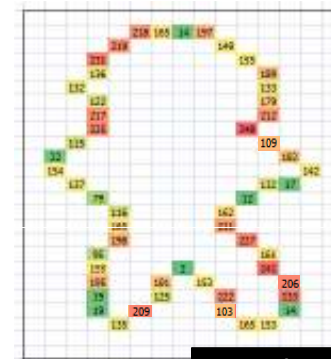


Bin of Direction

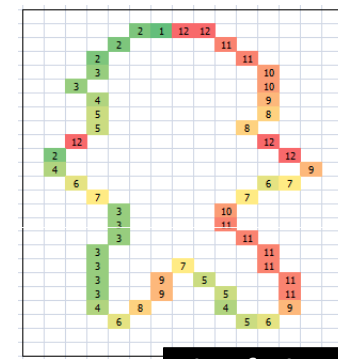


261458 & 261753 Computer Vision

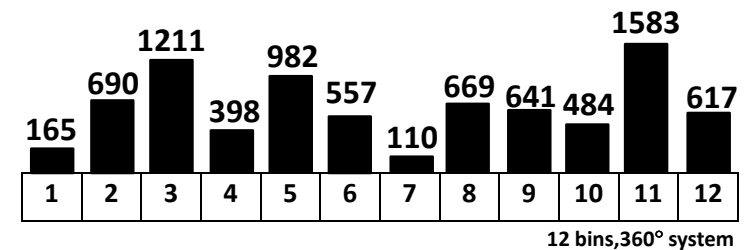
#4



Magnitude



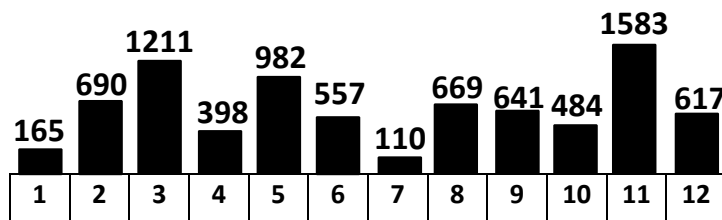
Bin of Direction



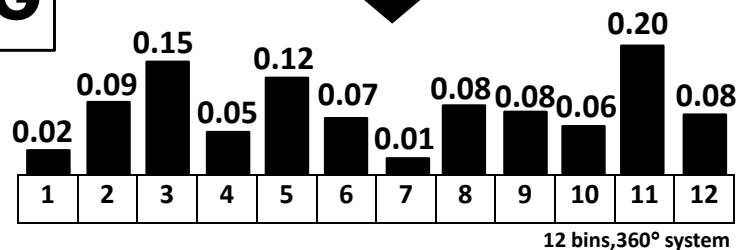
261458 & 261753 Computer Vision

#4

STEP 6: Normalization



HOG



Divide by 8107
(165+690+1211+398+982+557+110+669
+641+484+1583+617)

261458 & 261753 Computer Vision

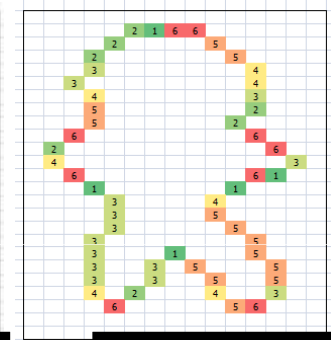
#4



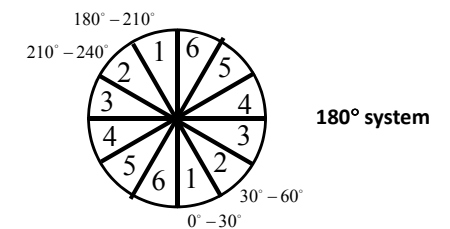
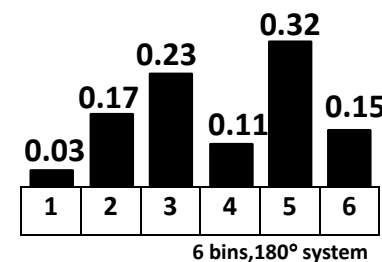
Magnitude



Direction



Bin of Direction

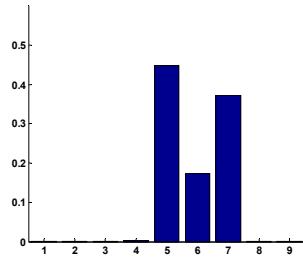
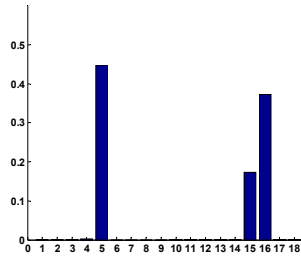
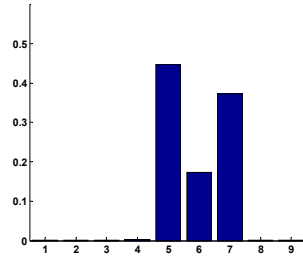
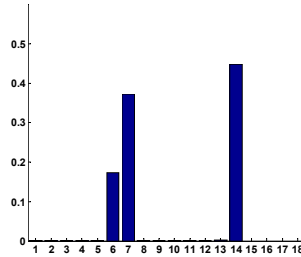


261458 & 261753 Computer Vision

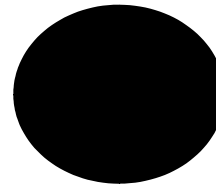
#4

360° system
(18 bins)

180° system
(9 bins)



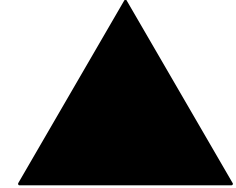
1



2

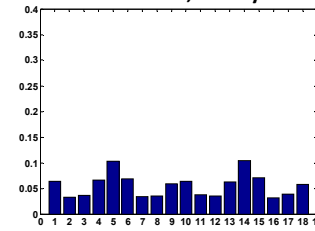


3



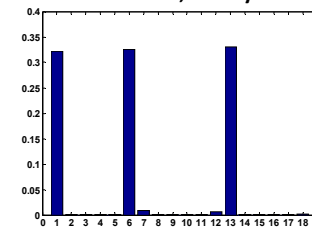
A

18 bins, 360° system



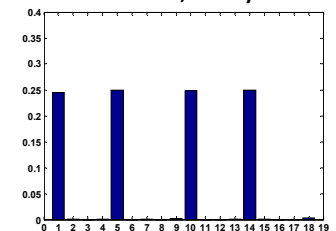
B

18 bins, 360° system

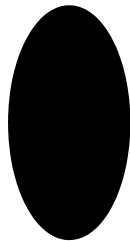


C

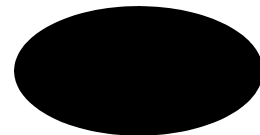
18 bins, 360° system



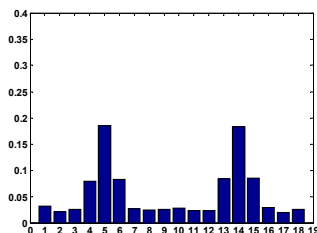
1



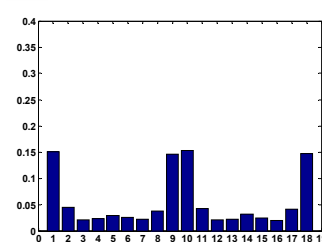
2



A

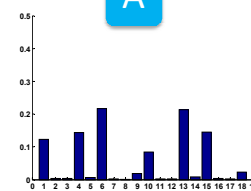


B

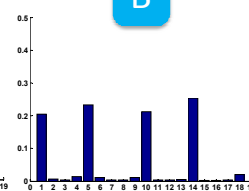


FAIL

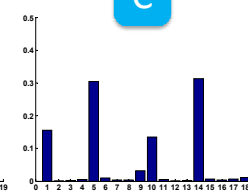
A



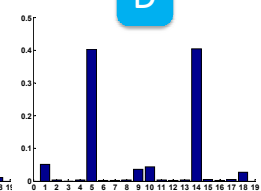
B



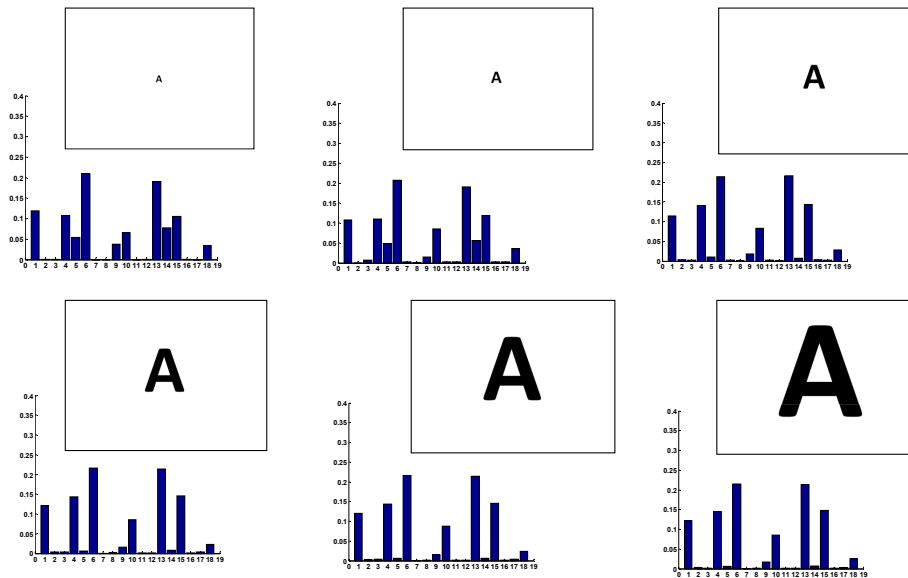
C



D



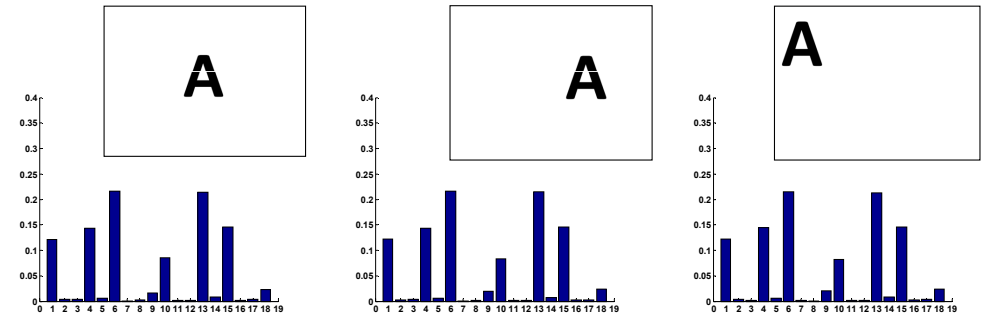
Scaling Invariant?



261458 & 261753 Computer Vision

#4

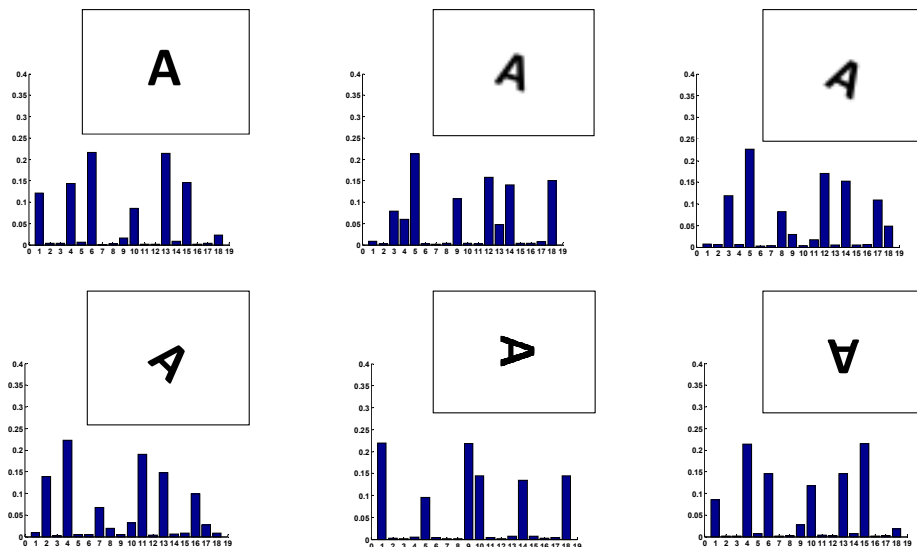
Translation Invariant?



261458 & 261753 Computer Vision

#4

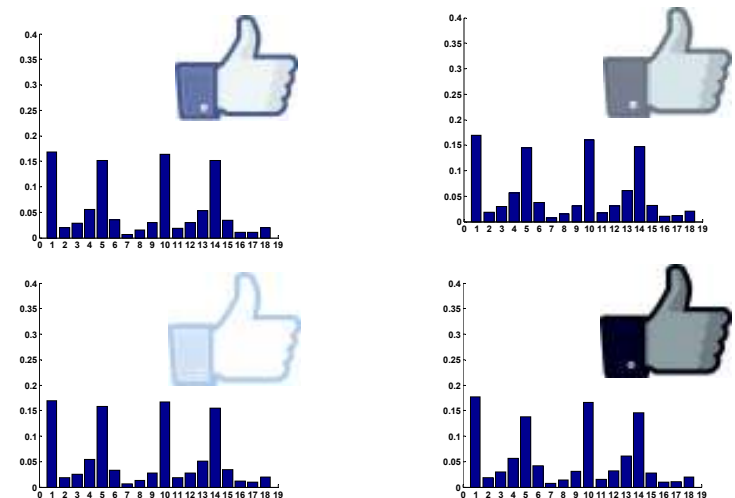
Rotation Invariant?



261458 & 261753 Computer Vision

#4

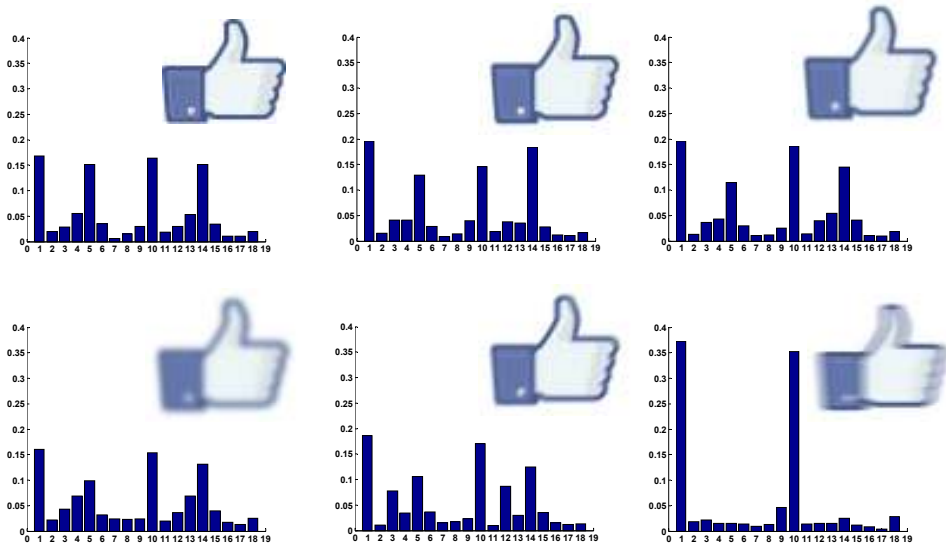
Robust to Intensity/Contrast variation?



261458 & 261753 Computer Vision

#4

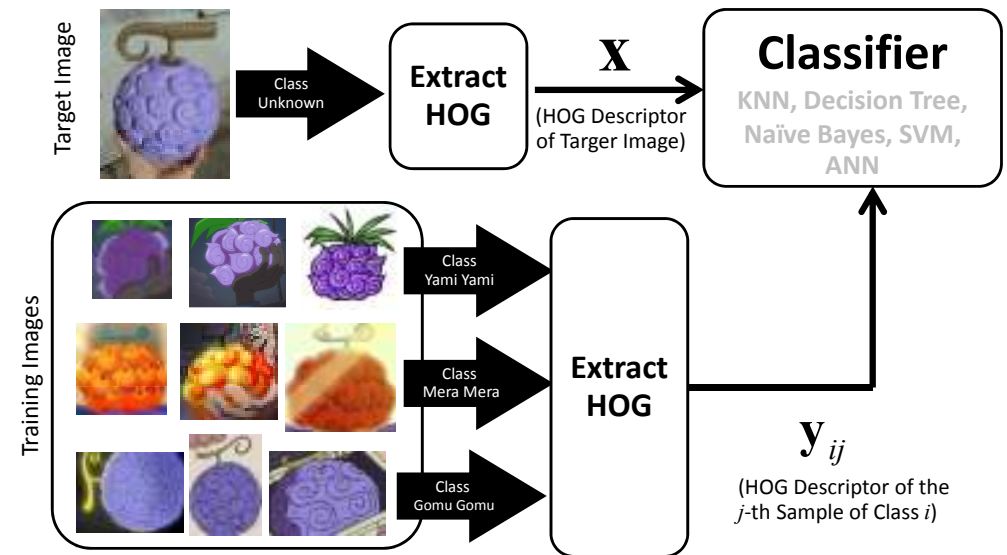
Robust to Blur?



261458 & 261753 Computer Vision

#4

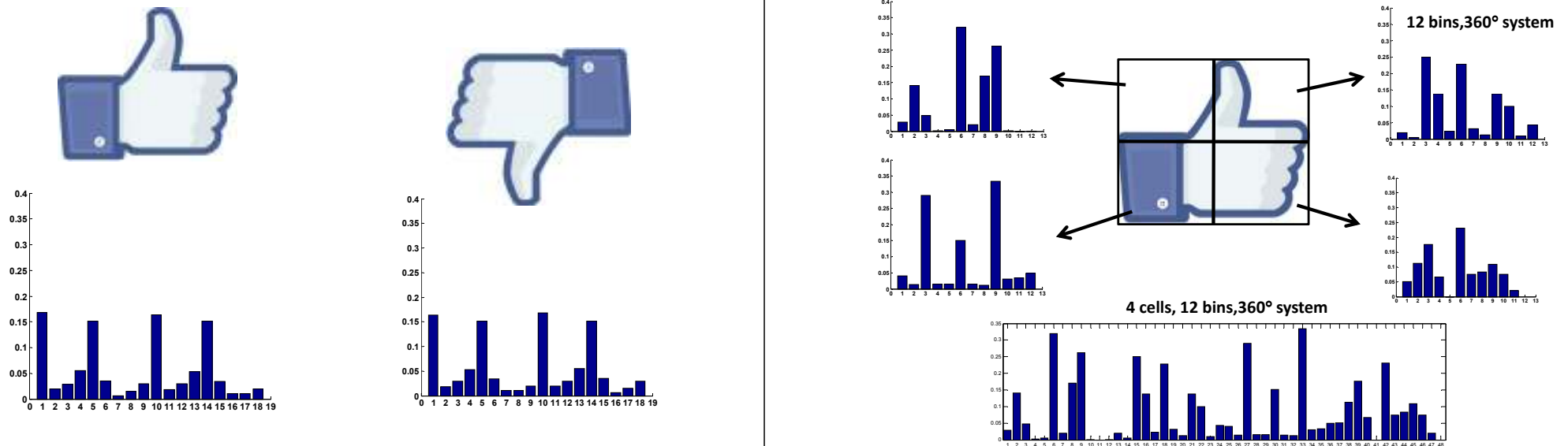
Basic Object Recognition by HOG



261458 & 261753 Computer Vision

#4

Segment Image into Cells



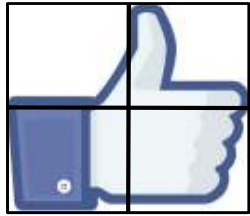
261458 & 261753 Computer Vision

#4

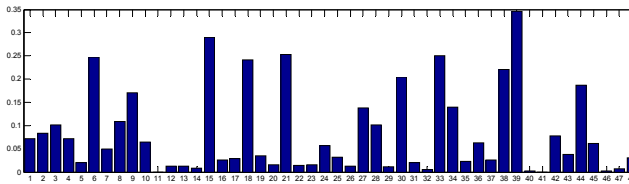
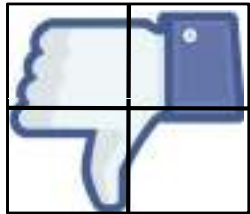
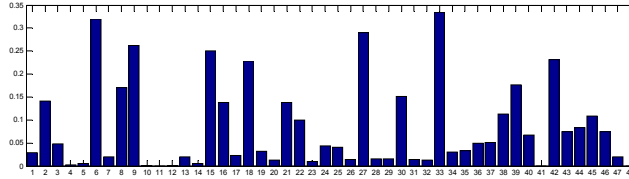
261458 & 261753 Computer Vision

#4

Segment Image into Cells



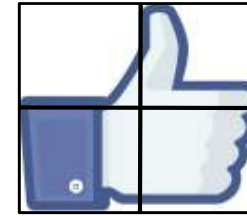
4 cells, 12 bins, 360° system



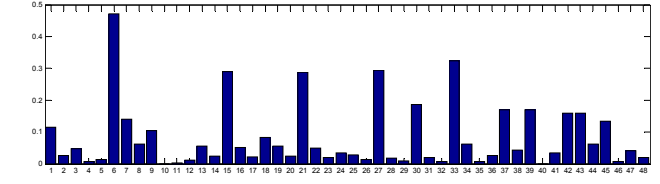
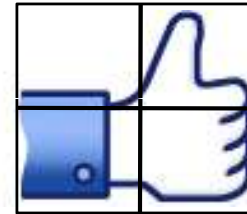
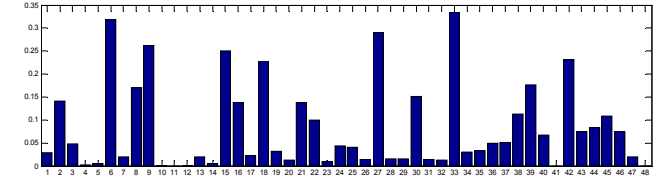
261458 & 261753 Computer Vision

#4

Segment Image into Cells



4 cells, 12 bins, 360° system

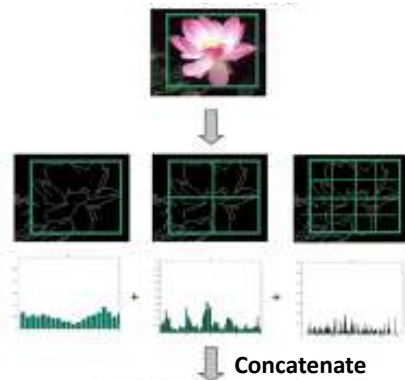


261458 & 261753 Computer Vision

#4

PHOG

Pyramid Histogram of Oriented Gradients



Concatenate

PHOG Descriptor

"Representing shape with a spatial pyramid kernel"

A.Bosch, A. Zisserman and X.Munoz

ACM, 2007

<http://www.robots.ox.ac.uk/~vgg/research/caltech/phog.html>

261458 & 261753 Computer Vision

#4

PHOG

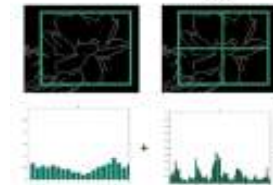
Pyramid Histogram of Oriented Gradients

Level = 0



PHOG Level 0 = HOG

Level = 1



PHOG Level 1 = [HOG & HOG of 4 cells]

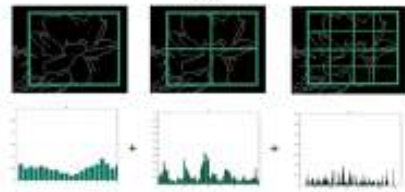
261458 & 261753 Computer Vision

#4

PHOG

Pyramid Histogram of Oriented Gradients

Level = 2



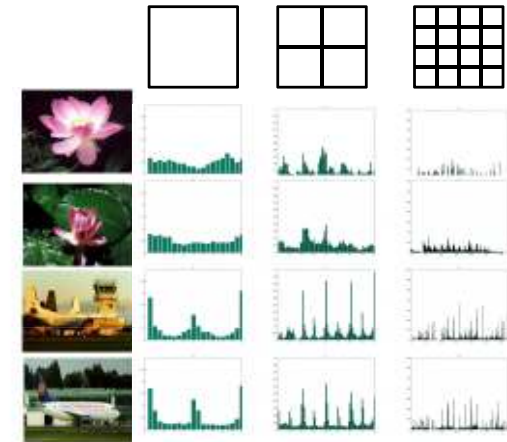
PHOG Level 2 = [PHOG Level 1 & HOG of 16 cells]

261458 & 261753 Computer Vision

#4

PHOG

Pyramid Histogram of Oriented Gradients

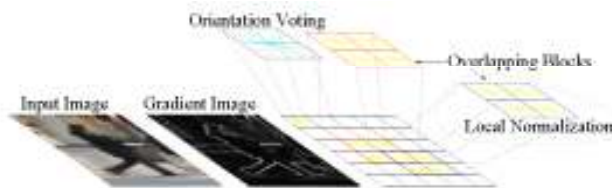


<http://eprints.pascal-network.org/archive/00003009/01/bosch07.pdf>

261458 & 261753 Computer Vision

#4

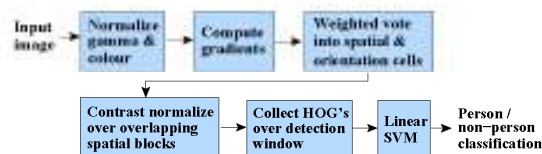
Object Detection Framework using HOG



"Histograms of Oriented Gradients for Human Detection"

N. Dalal and B. Triggs

CVPR 2005



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

261458 & 261753 Computer Vision

#4

Object Detection Framework using HOG



Positives



Negatives

[1] Prepare training set

-> Positive Images

(Image of the target objects

-> Negative Images

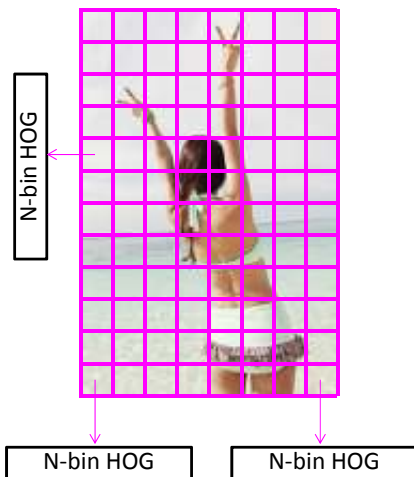
(Scene images without target objects for random window selection)

(Resize every sample to obtain the same size)

261458 & 261753 Computer Vision

#4

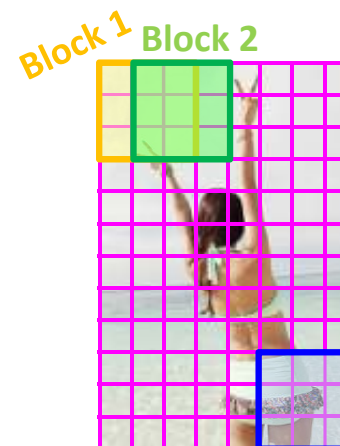
Object Detection Framework using HOG



[2] For each training samples, divide images into **cells**.

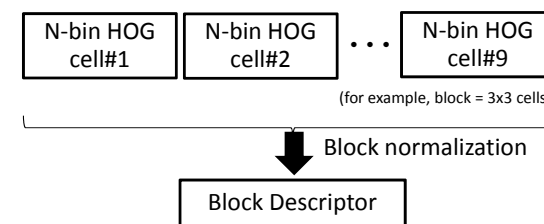
[3] Compute HOG for each cells

Object Detection Framework using HOG

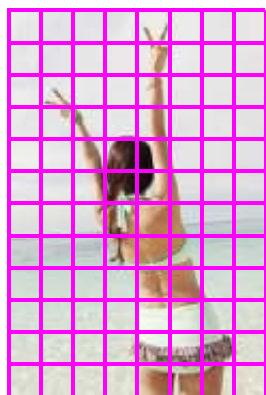


[4] Group cells into **blocks**
(better to have overlap)

[5] Concatenate HOG of every cell in that block and normalize it.



Object Detection Framework using HOG



[6] Concatenate every block descriptor to obtain the final descriptor for the image.

For example:
Image size = 64x128 pixels
Cell size = 8x8 pixels => #Cell = 8x16 cells
Block size = 2x2 cells (50% overlap)
=> #Block = 7x15 blocks
HOG Bin = 9 bin/cell
=> Length of Block descriptor = 9x2x2 = 36
=> Length of Final descriptor = 36x7x15 = 3780

Object Detection Framework using HOG



Positives

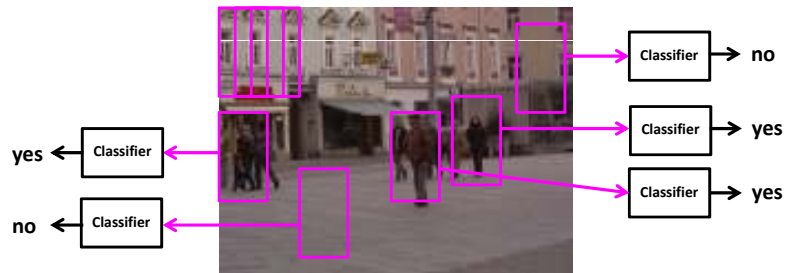
[7] Compute final descriptor for every positives image and random windows of negative images



Negatives

[8] Use descriptors obtained from training set to train a binary **SVM classifier**.

Object Detection Framework using HOG



[9] Apply sliding window to scan over the target image and use classifier to determine whether that window is the desired object or not.