EE655: Computer Vision & Deep Learning

Lecture 04

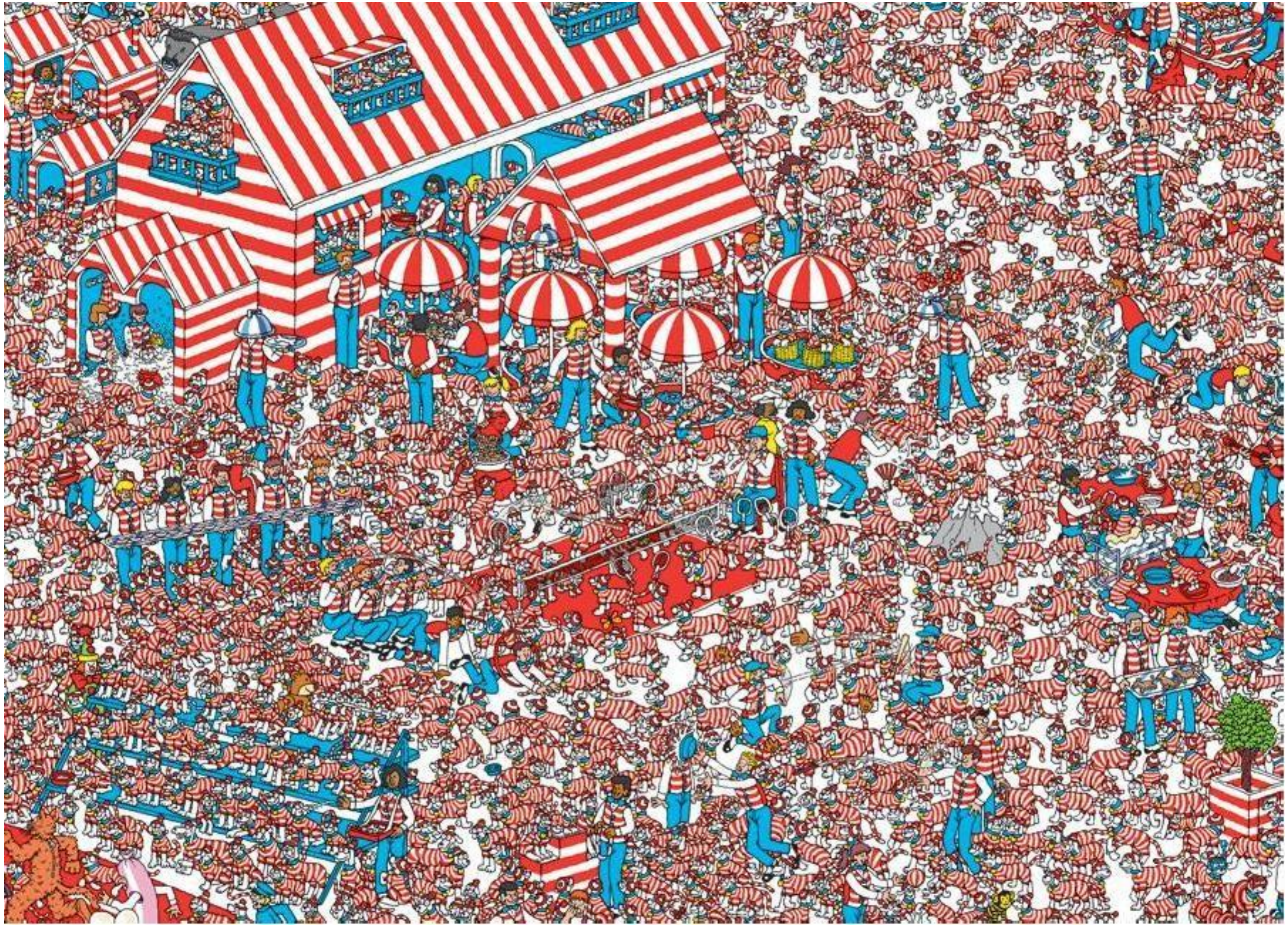
Koteswar Rao Jerripothula, PhD
Department of Electrical Engineering
IIT Kanpur

Lecture Outline

Histogram of Oriented Gradients (HOG)

Feature Extraction (in General)

Local Binary Patterns (LBP)



Can you spot where is this sub-image in the full image?

Can we do it programmatically?

11	13	34
34	24	21
64	55	64
68	45	25
80	31	45

CORRELATION == DOT PRODUCT

20	35	21
53	22	23
22	34	32
22	56	55
45	11	13

$I =$

11
34
64
68
80
.
.
25
45

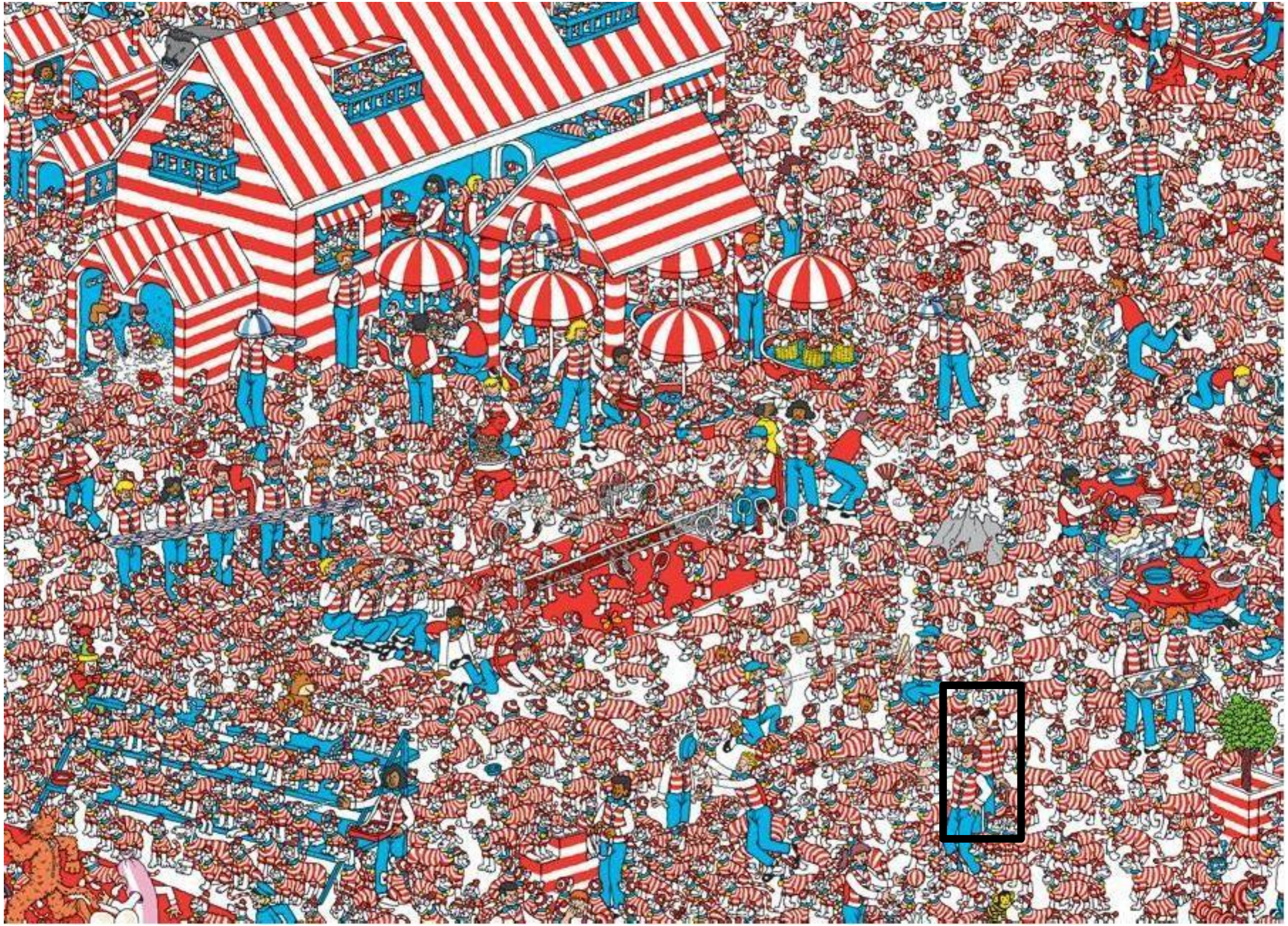
$$\frac{u \cdot v}{\|u\| \|v\|} = \cos\theta$$

Measure of similarity between normalized vectors

$J =$

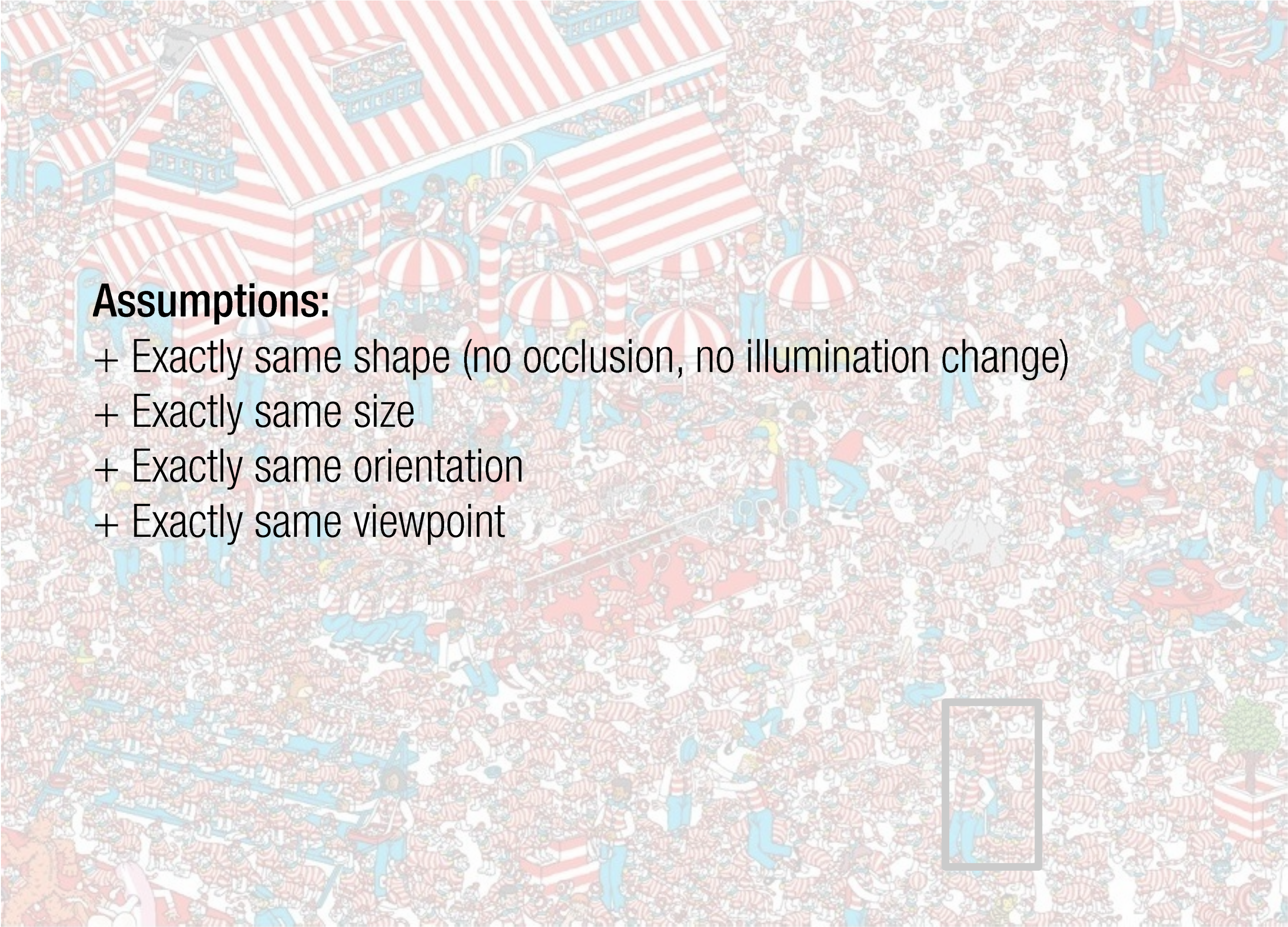
20
53
22
22
45
.
.
55
13

$Corr\Big[$  ,  $\Big]$



We will find the highest similarity value at the exact match

While correlation can be used to find similarity, there are some inherent assumptions involved



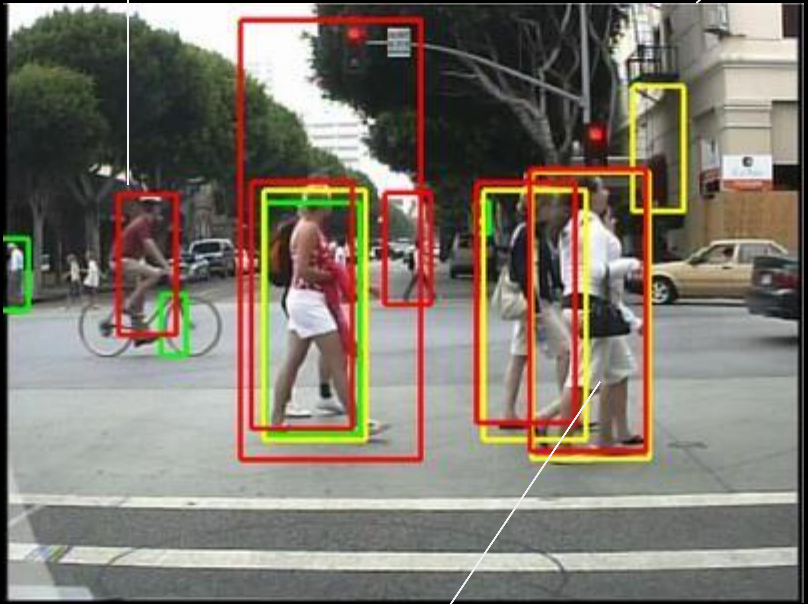
Assumptions:

+ Exactly same shape (no occlusion, no illumination change)
+ Exactly same size
+ Exactly same orientation
+ Exactly same viewpoint

Reality while detecting pedestrians

Different illumination

Views

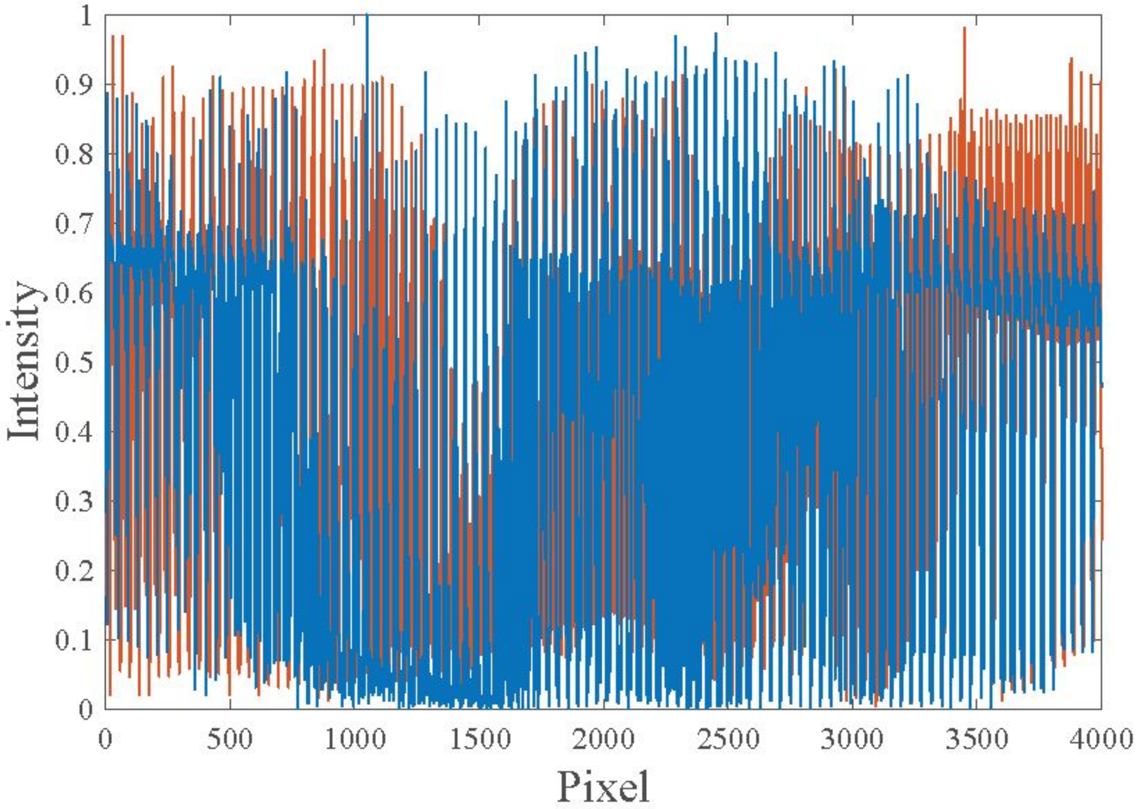


Occlussion

Various poses

Pixel Intensity Correlation

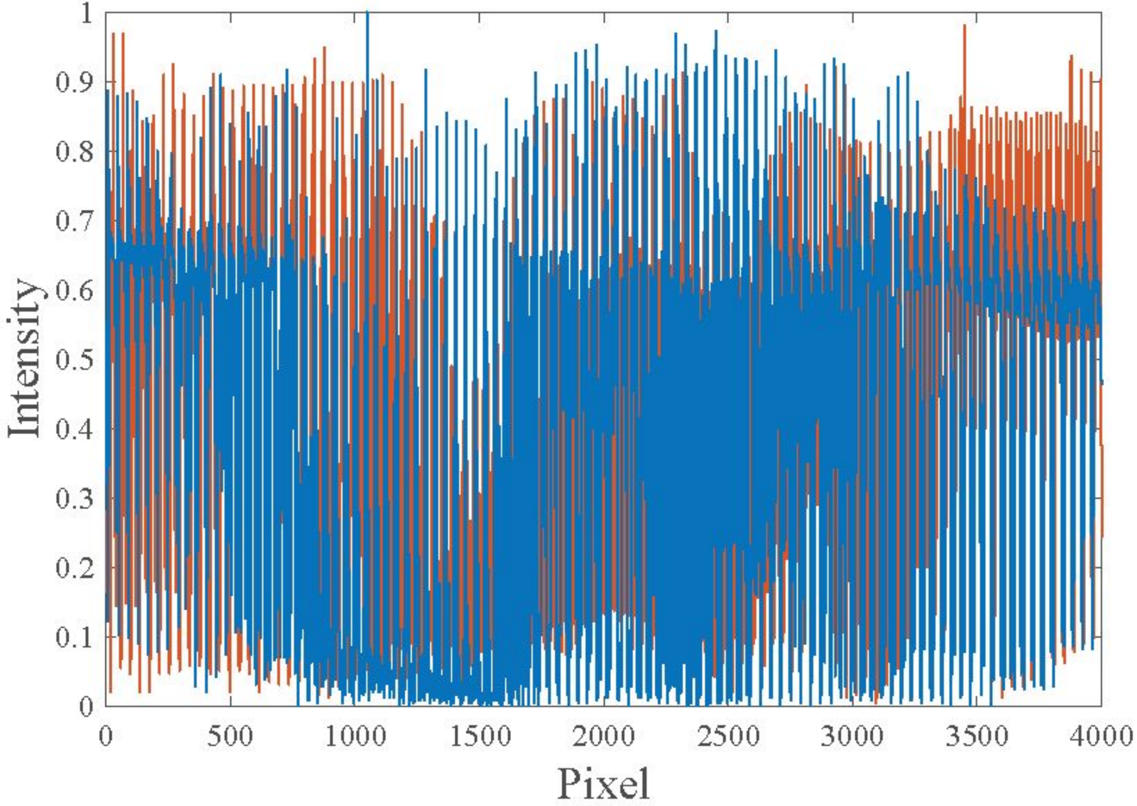
Corr = 0.24



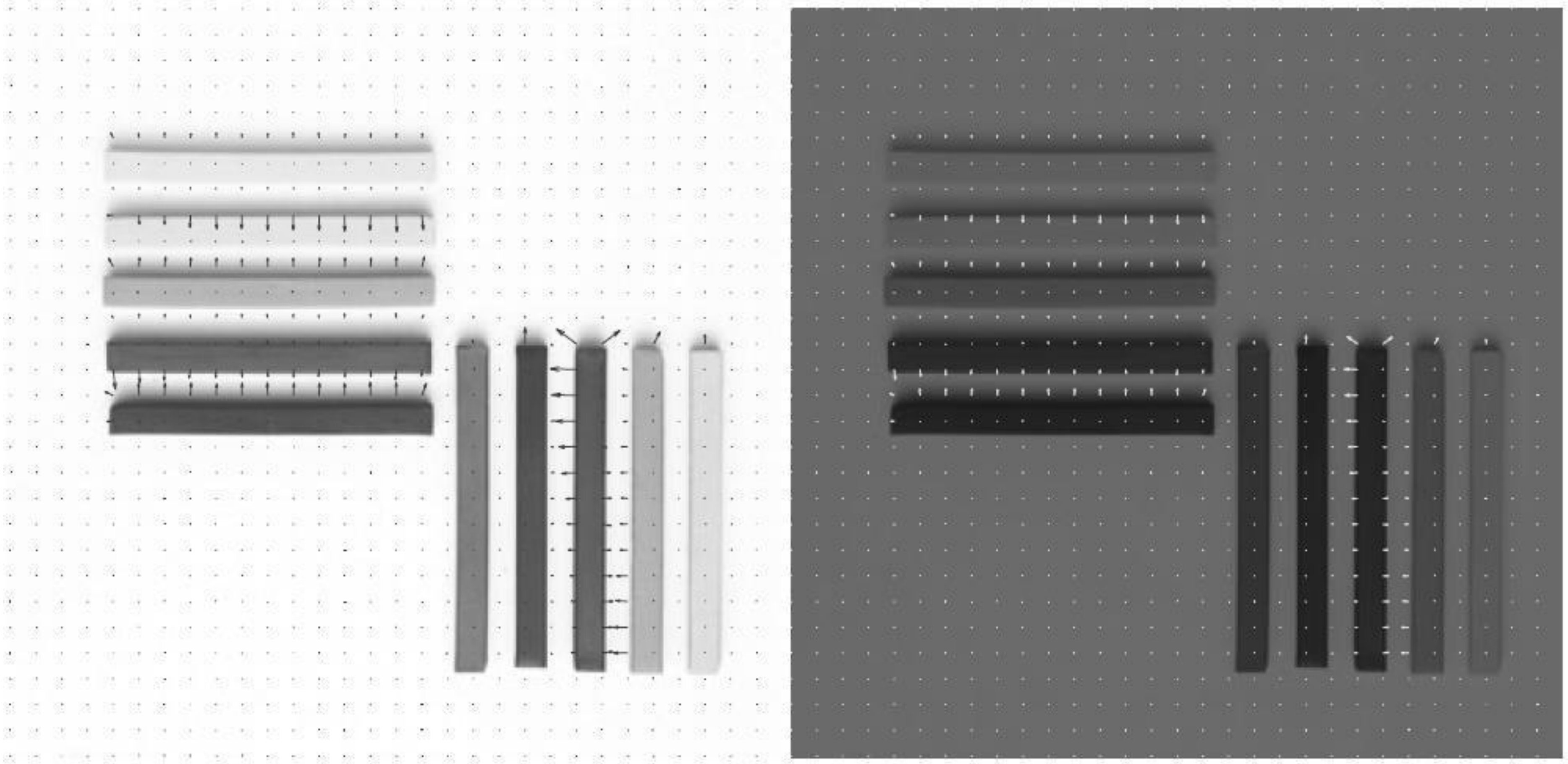
PIXEL INTENSITY CORRELATION

Corr = 0.24

Corr = 0.31



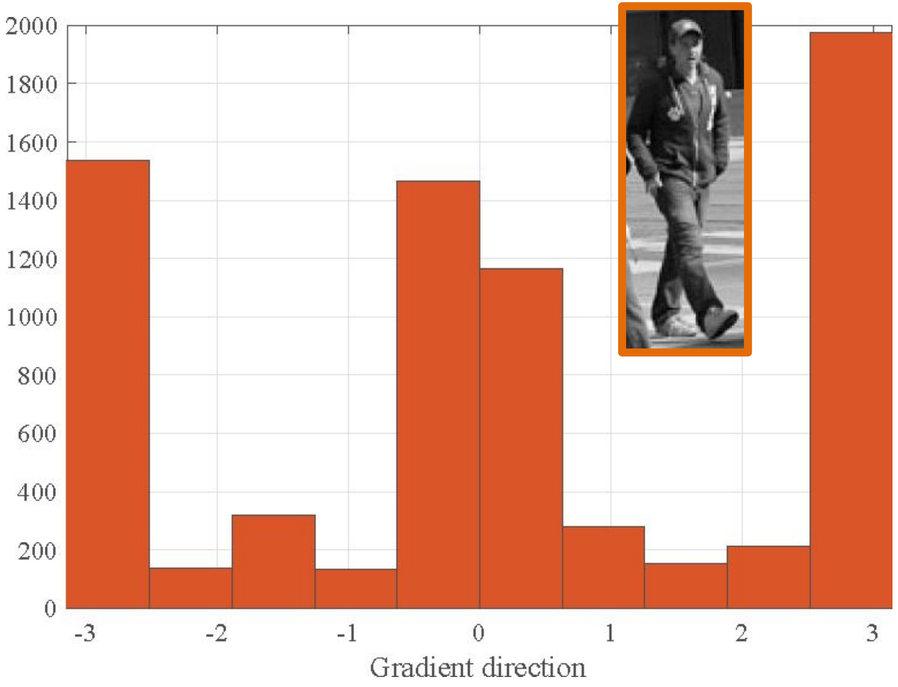
Recall: Illumination Invariant Gradient

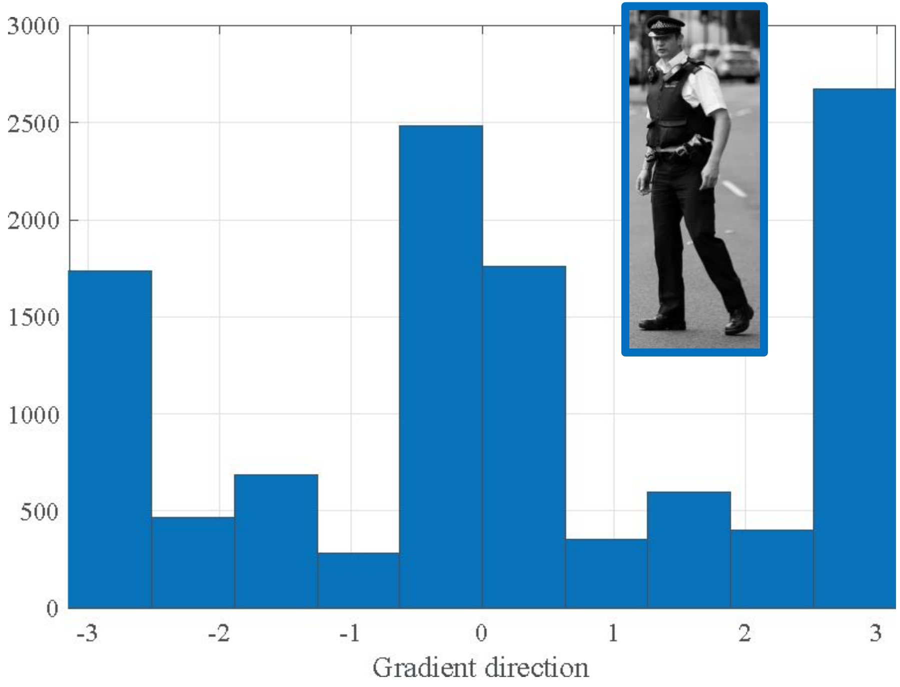


GRADIENT

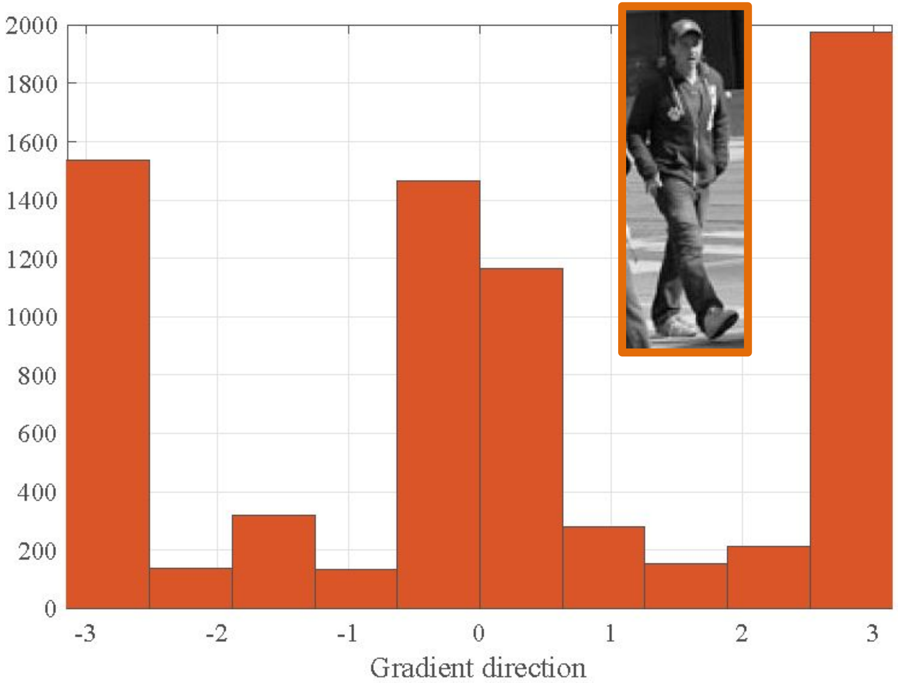
$\|\nabla I\|$ $\measuredangle \nabla I$ $\measuredangle \nabla I$ with mag thr. $\|\nabla J\|$ $\measuredangle \nabla J$ $\measuredangle \nabla J$ with mag thr.

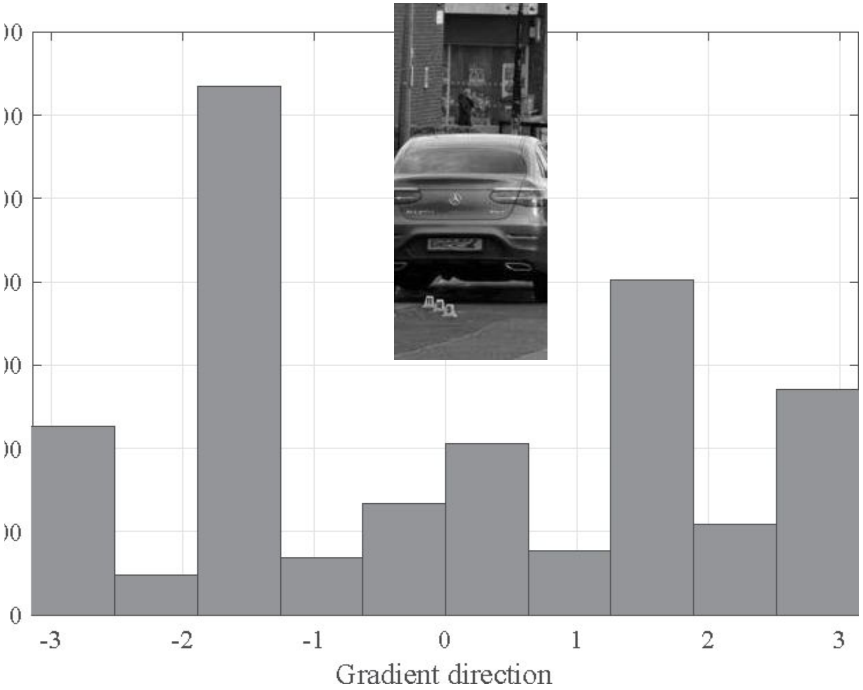
GRADIENT DISTRIBUTION (HISTOGRAM OF GRAD. DIRECTION)



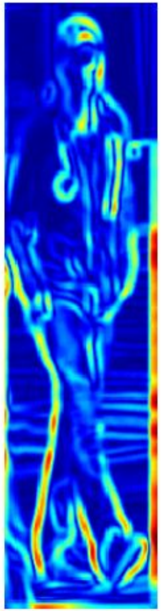


Gradient Distribution (Histogram of Grad. Direction)

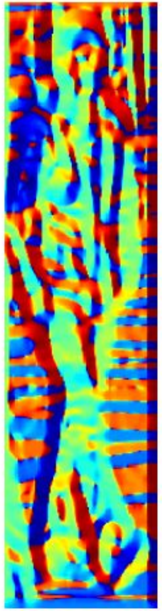




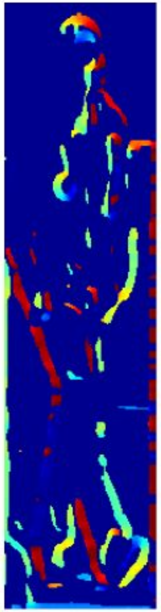
GLOBAL GRADIENTS



$\|\nabla I\|$

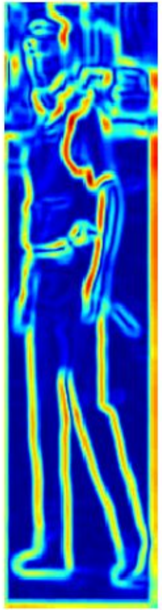


$\measuredangle \nabla I$

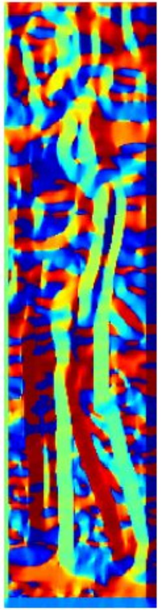


$\measuredangle \nabla I$

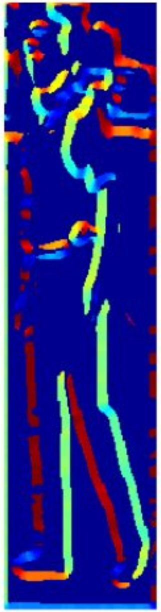
with mag thr.



$\|\nabla J\|$



$\measuredangle \nabla J$

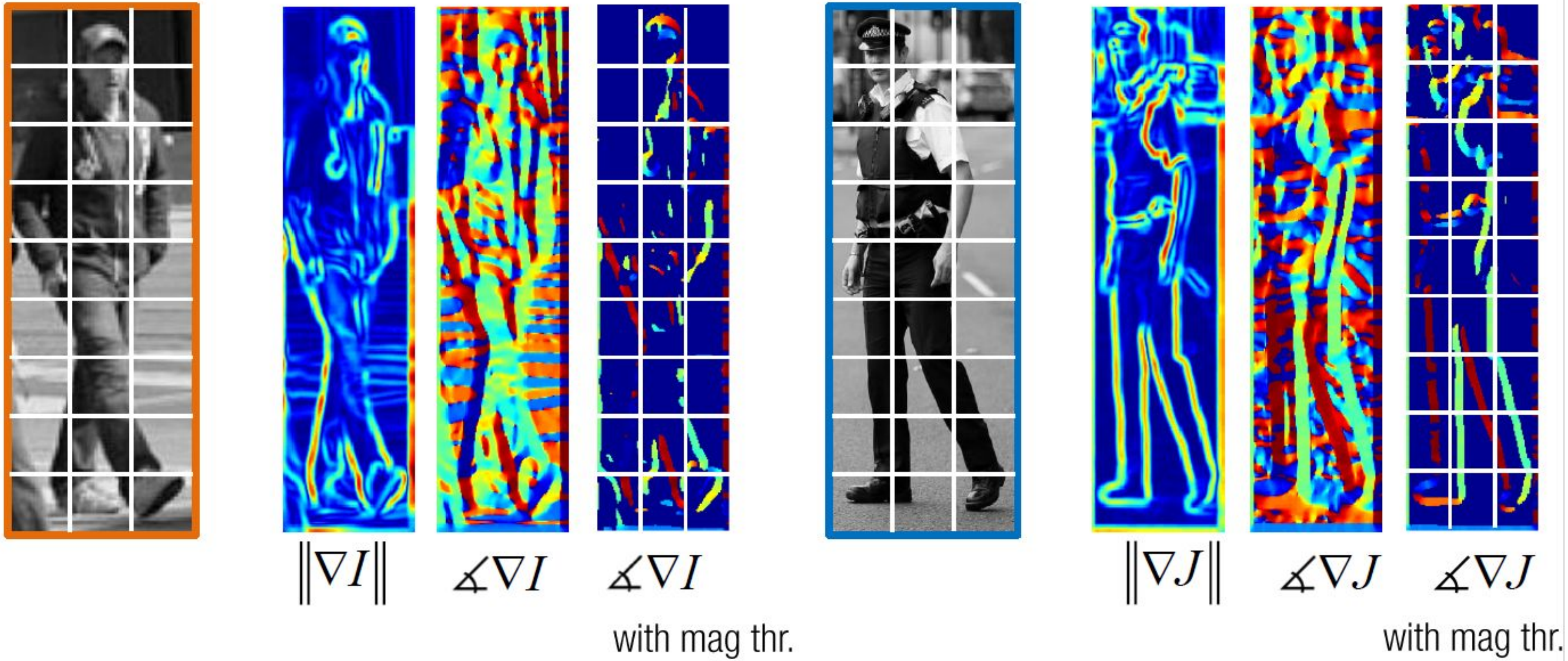


$\measuredangle \nabla J$

with mag thr.

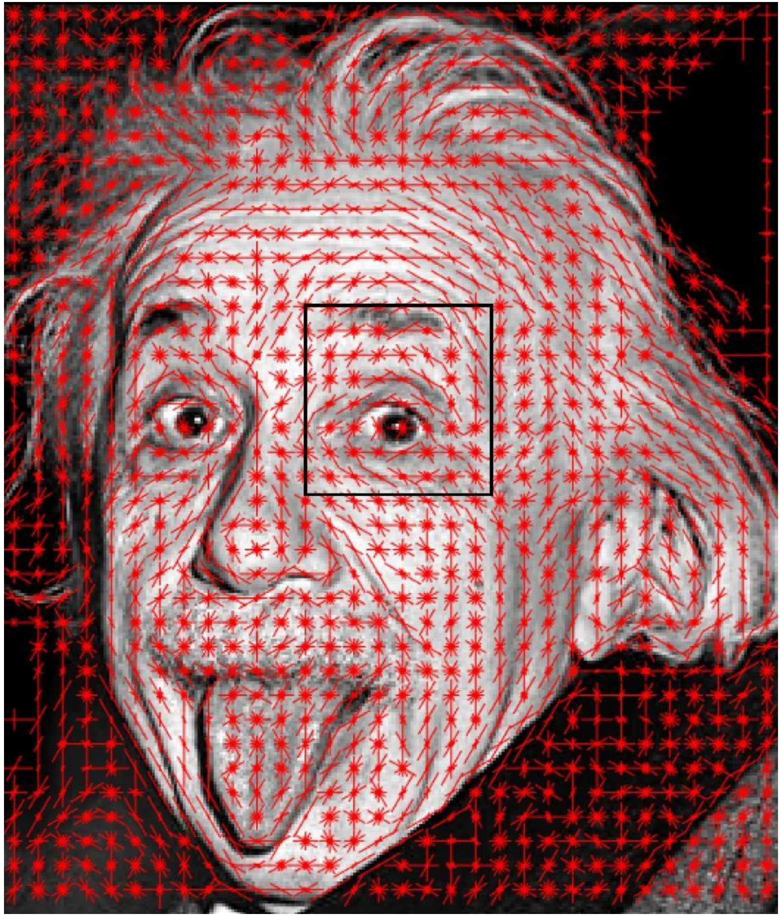
LOCAL GRADIENTS

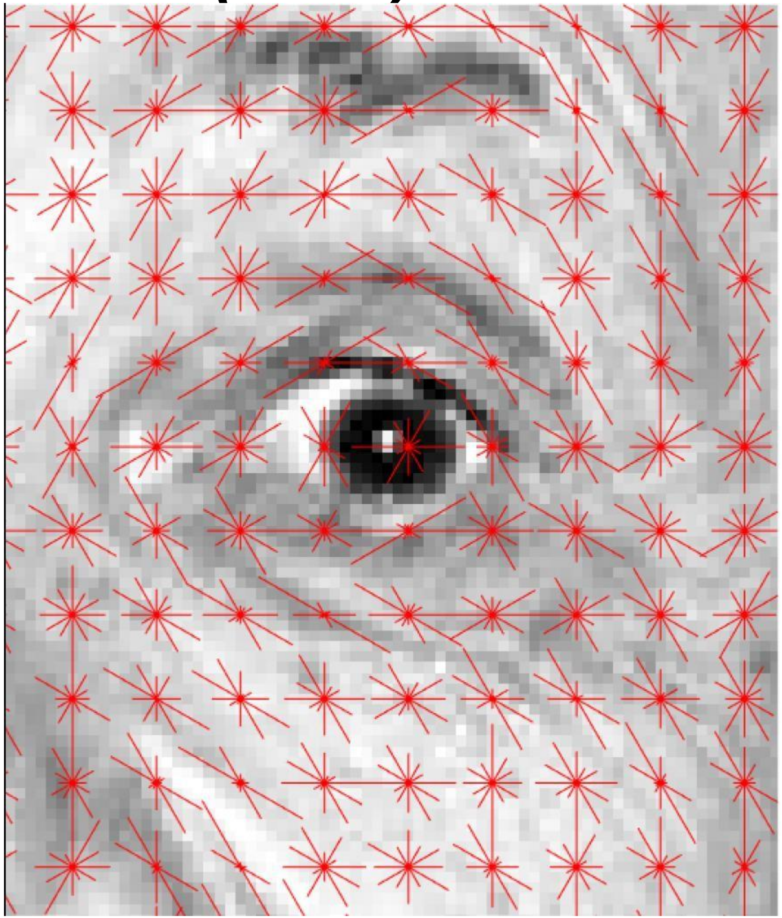
We can also do fine-grained analysis at cell (patch) level and concatenate the histograms



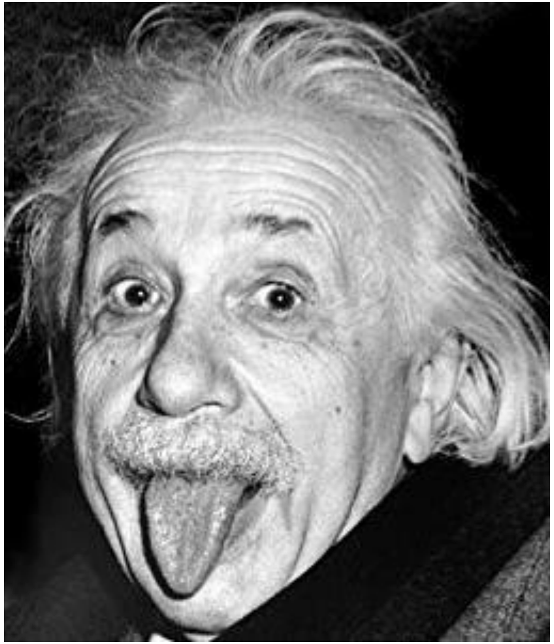
HISTOGRAM OF ORIENTED GRADIENTS (HOG)

Each cell (patch) has one HOG

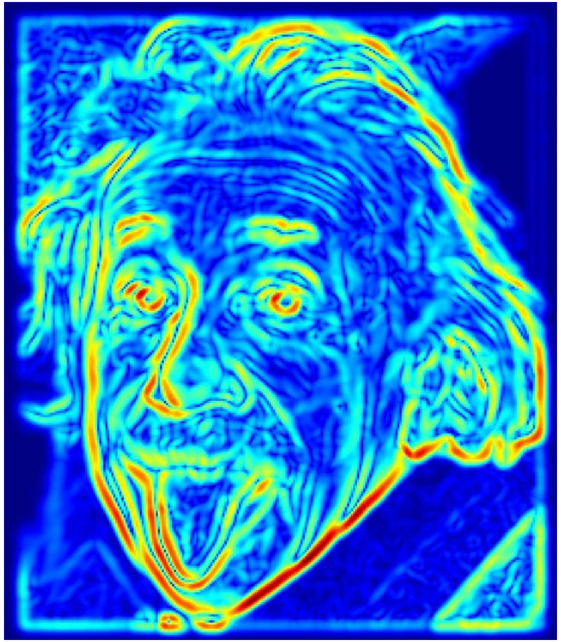




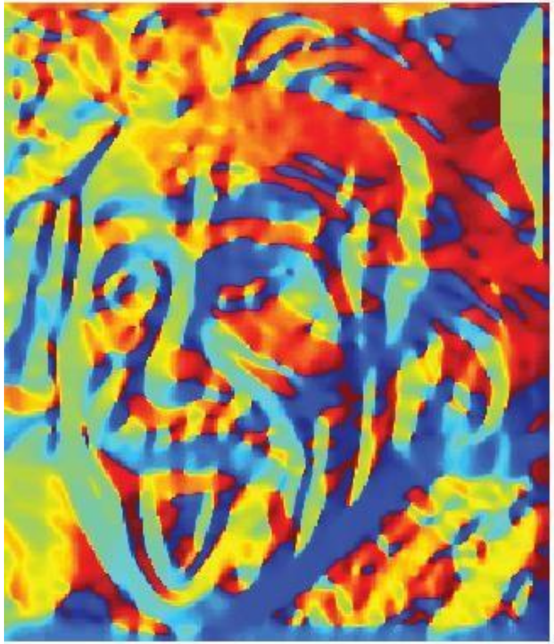
Gradient Image



I

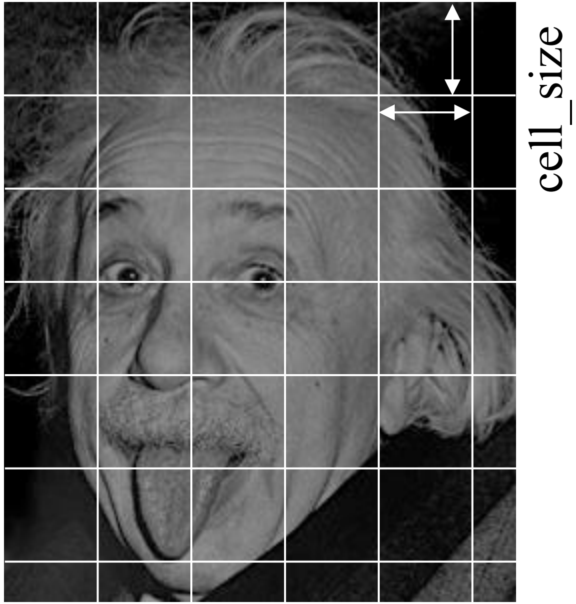


$\|\nabla I\|$

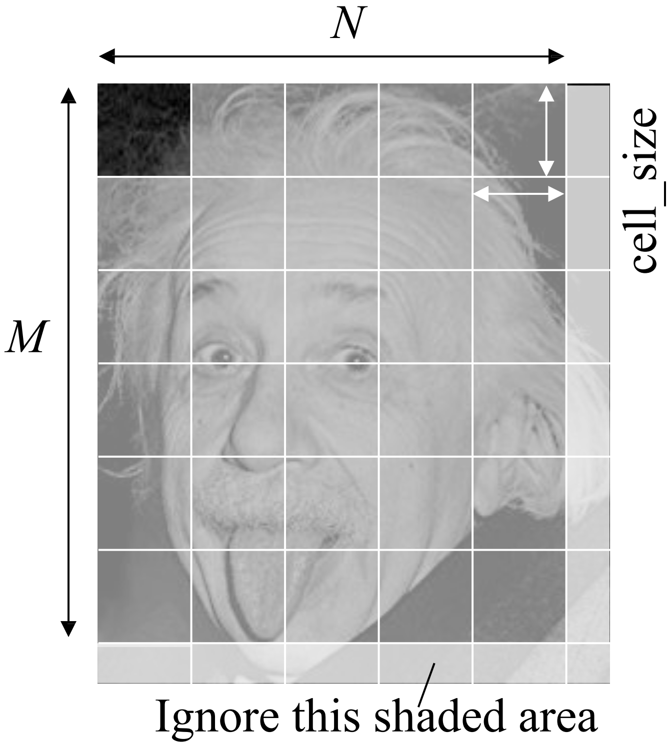


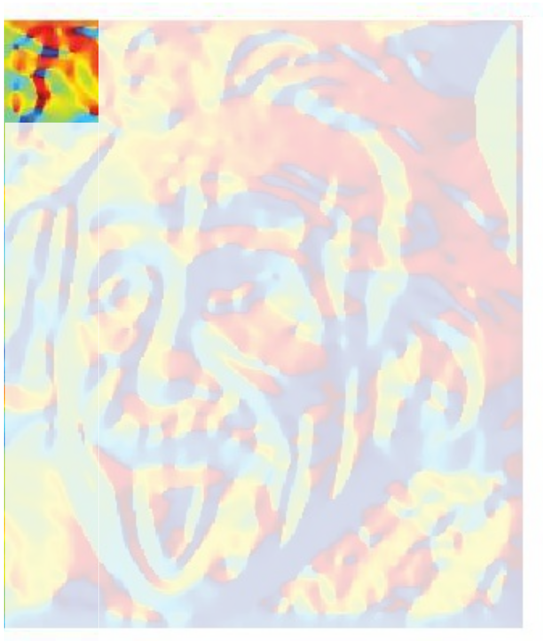
$\measuredangle \nabla I$

ORIENTATION BINNING



ORIENTATION BINNING

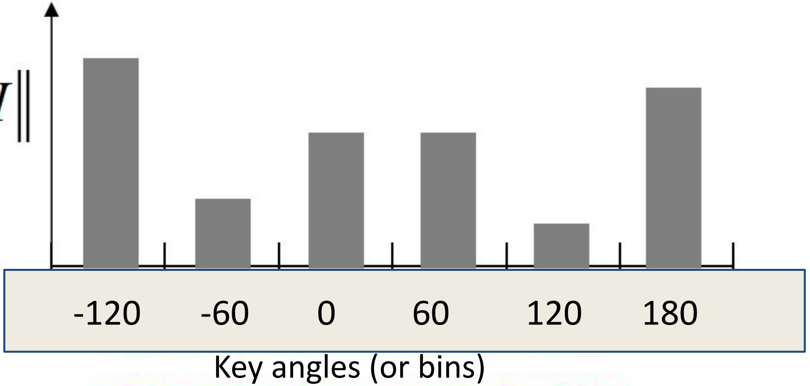


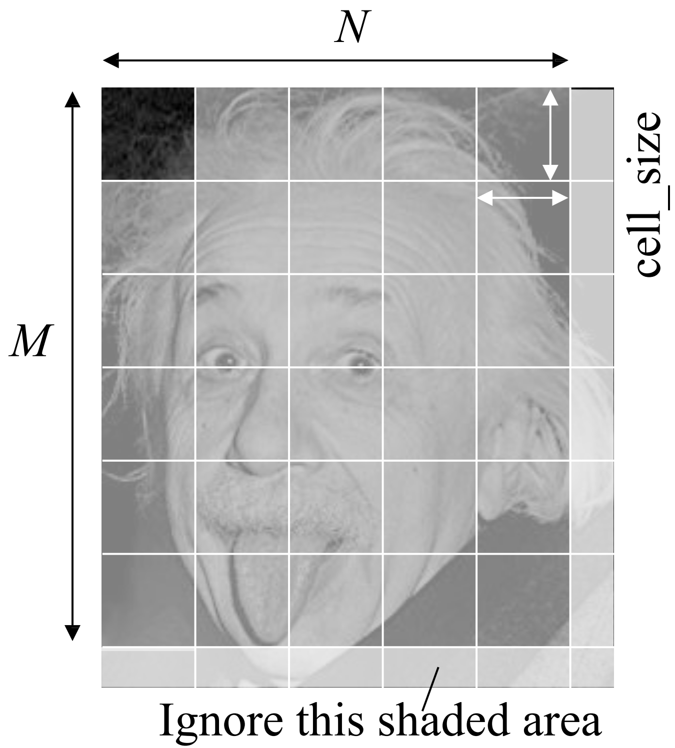


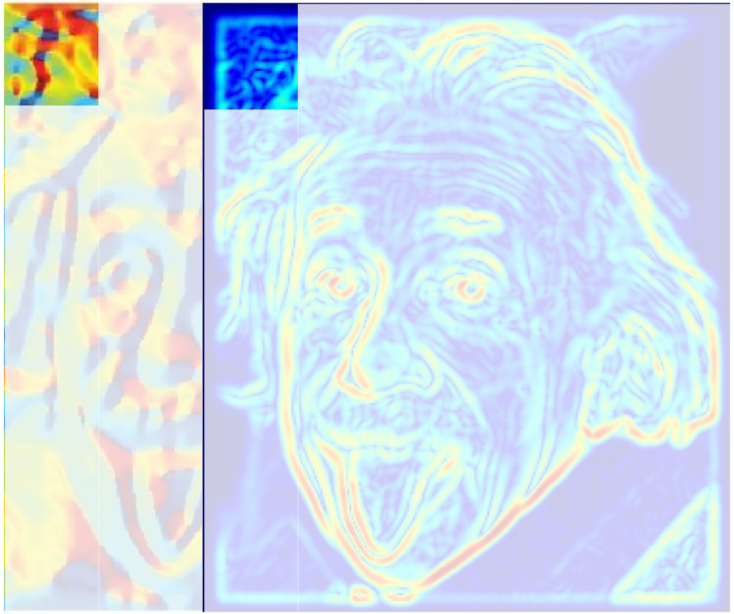
Orientation Binning

$$\sum \|\nabla I\|$$

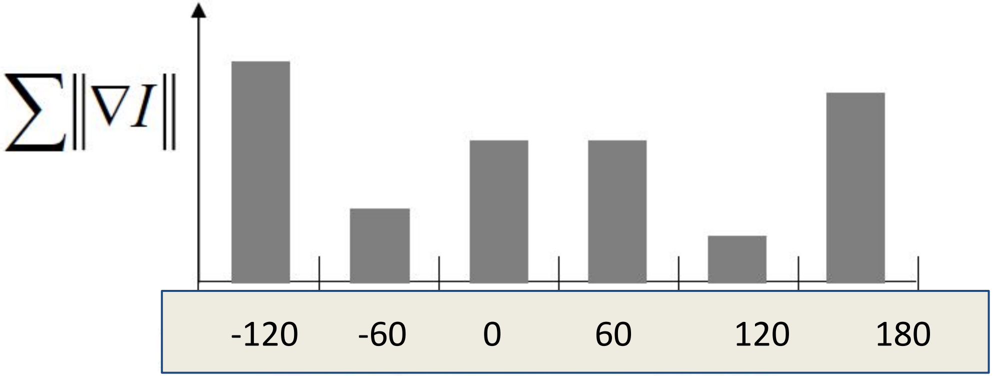
NOTE: Sum of Magnitudes, not frequency

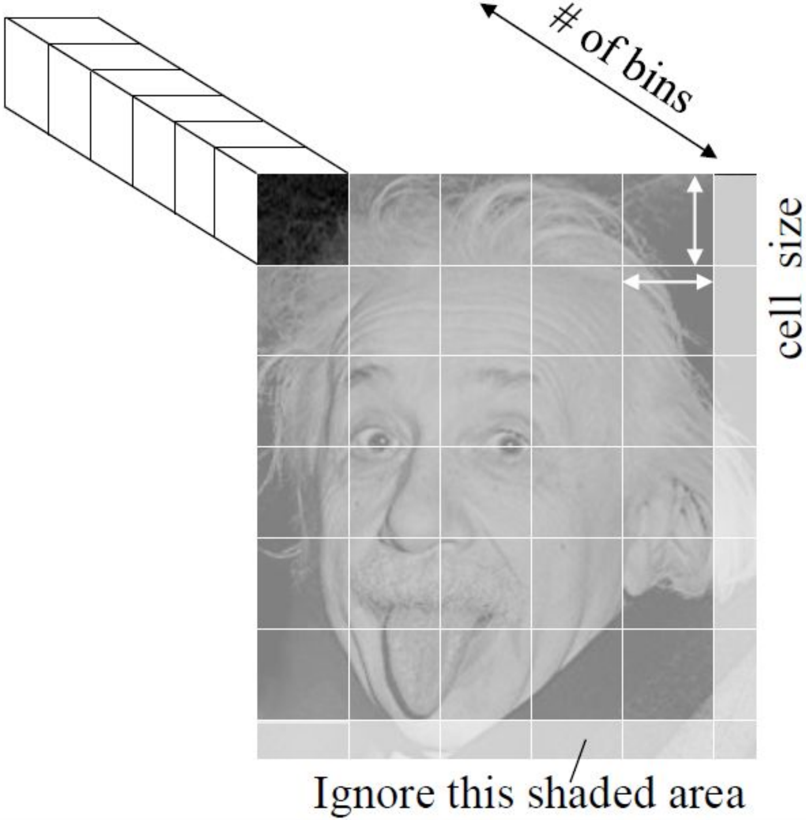


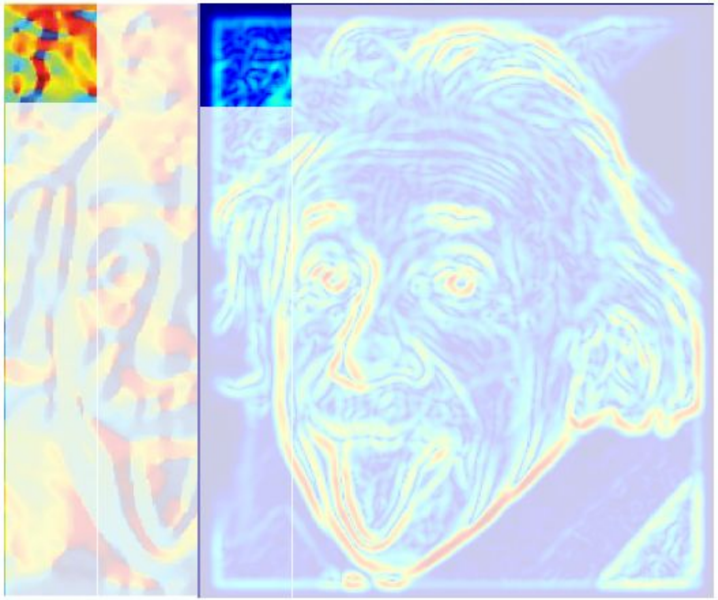




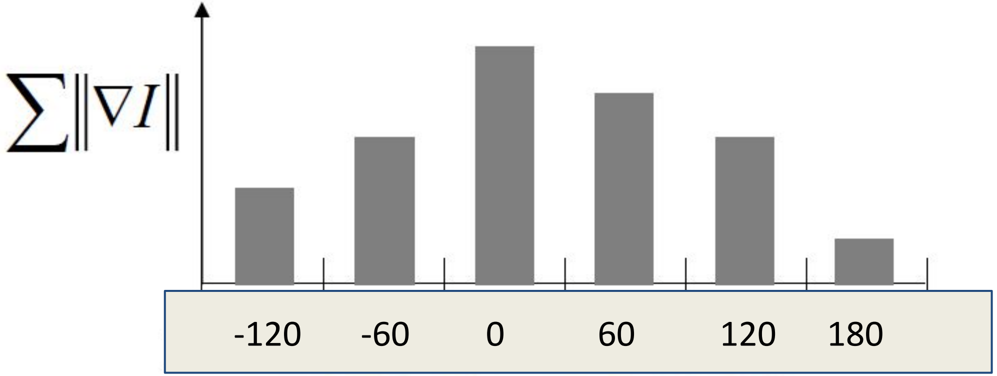
ORIENTATION BINNING

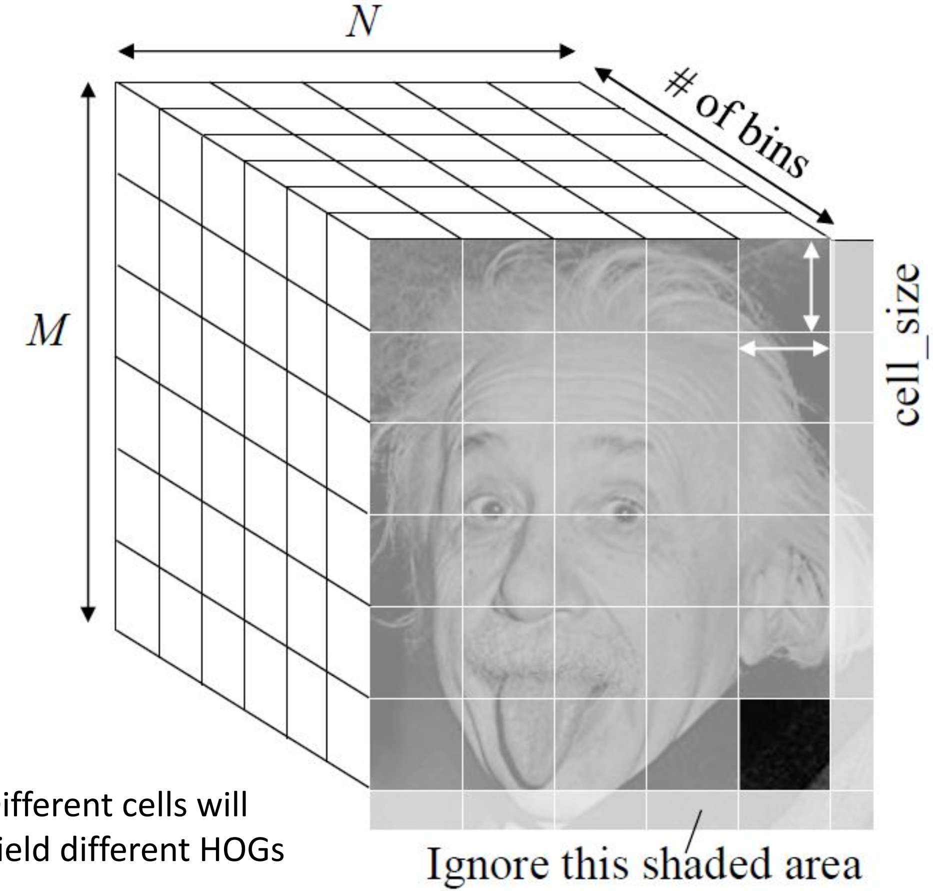


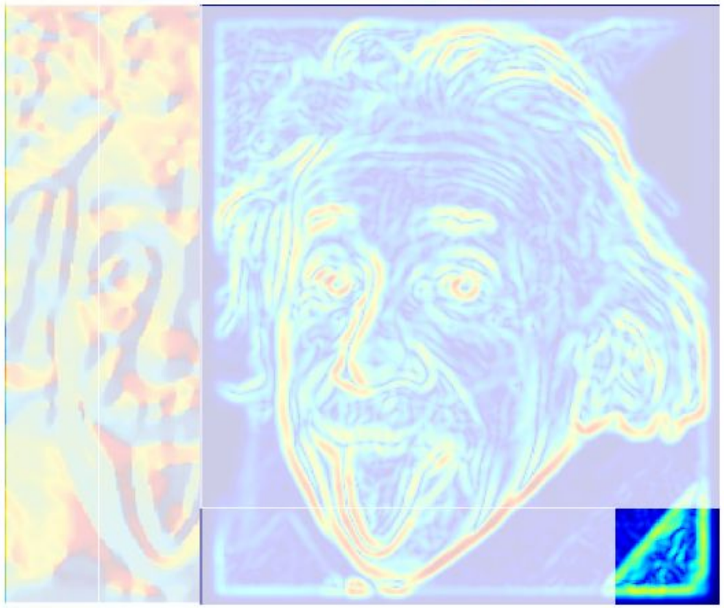




ORIENTATION BINNING

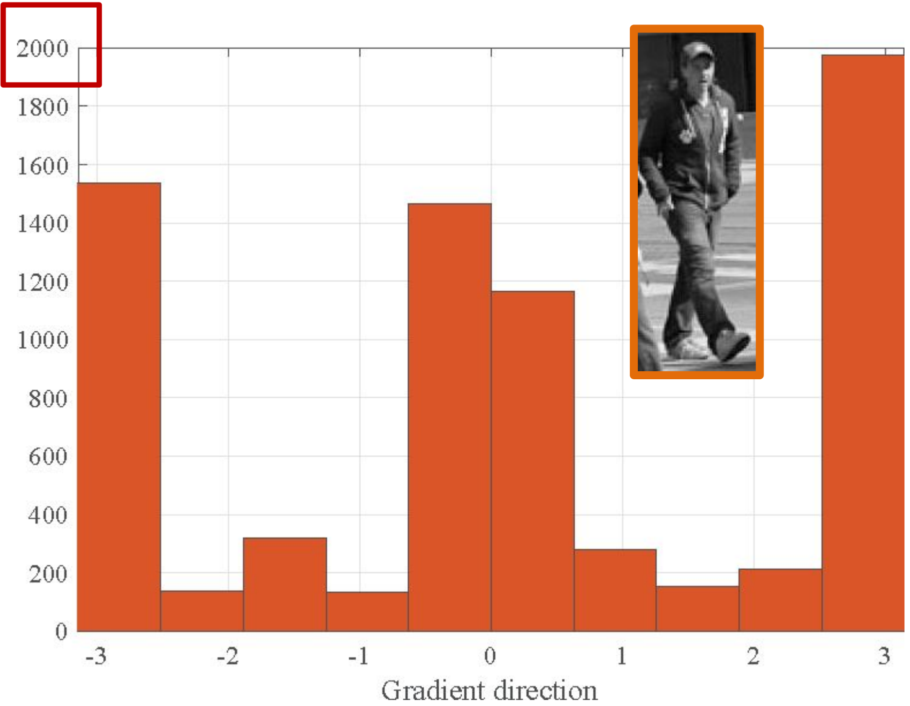


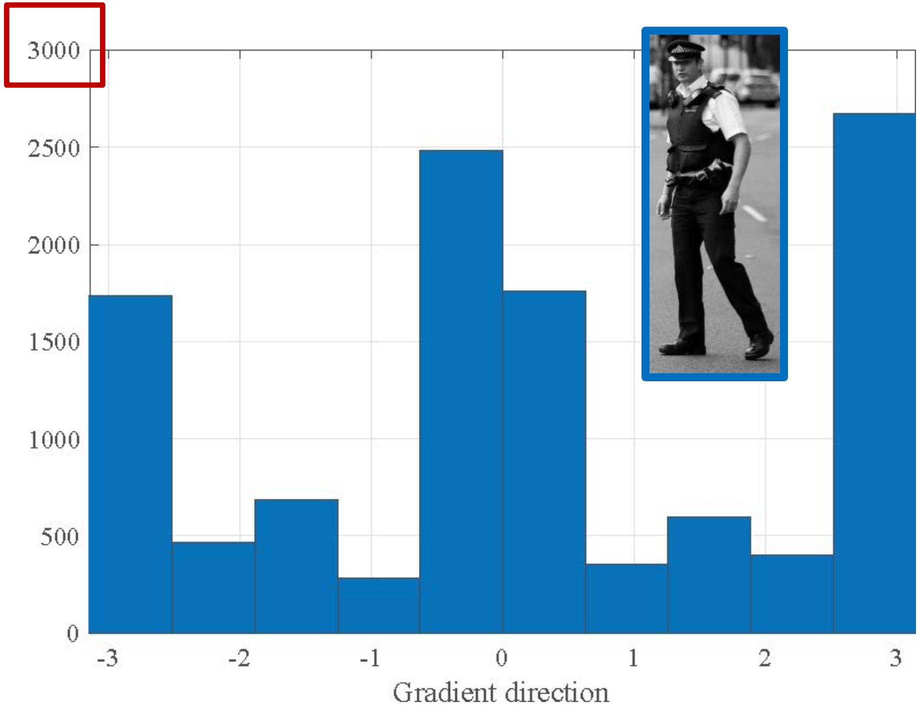




Different cells will
yield different HOGs

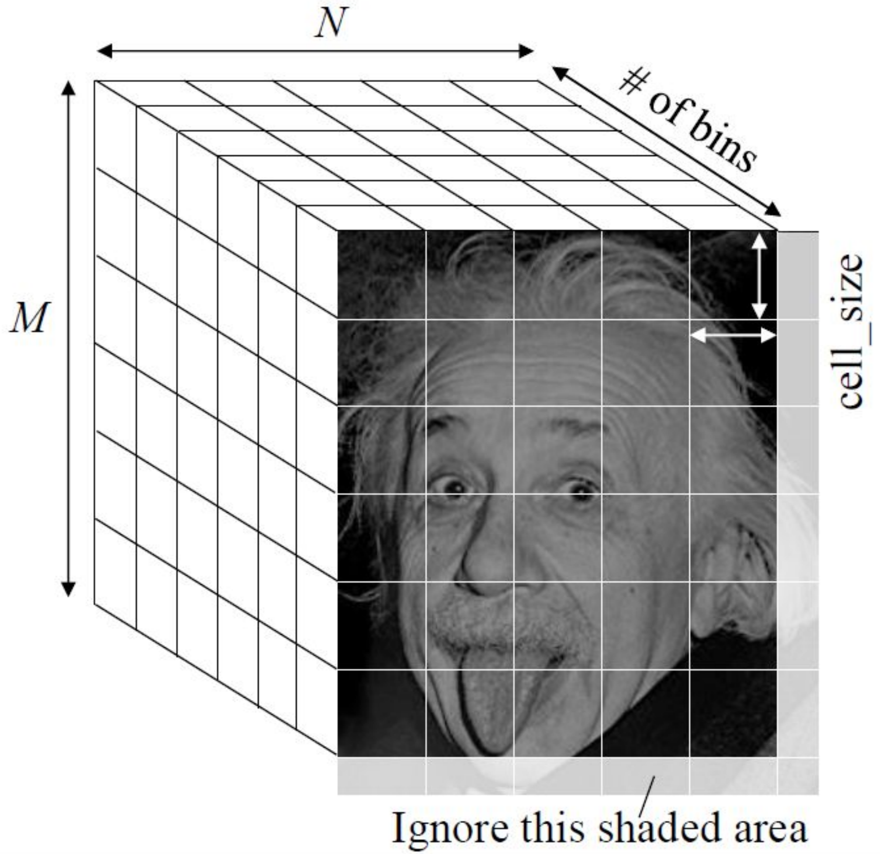
RECALL: SCALE

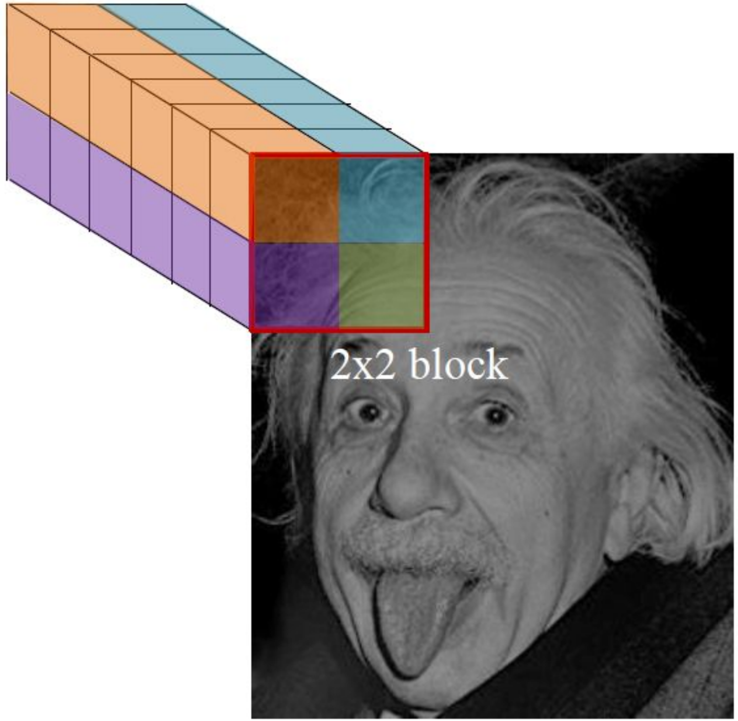




BLOCK NORMALIZATION

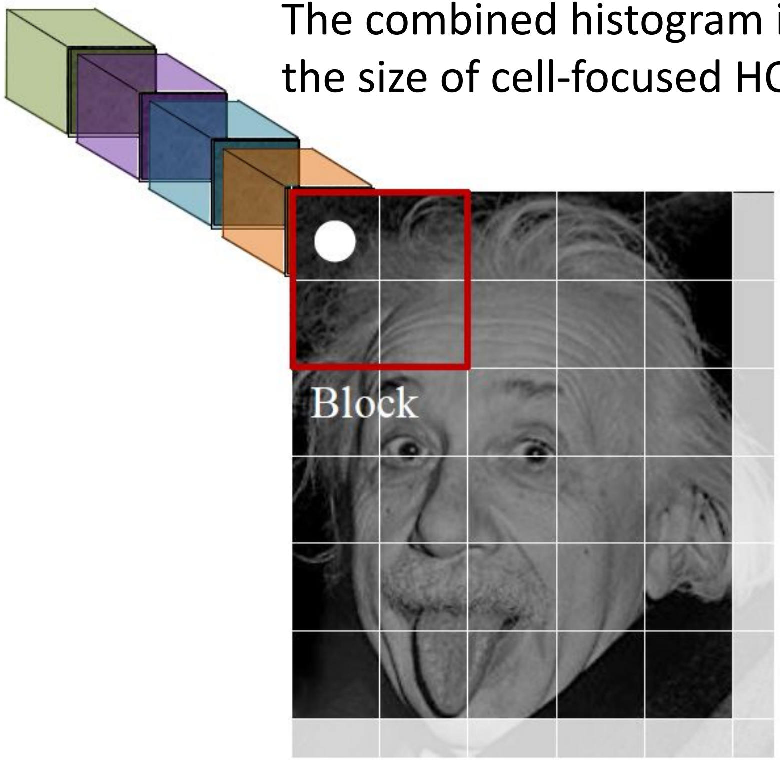
We will now include HOGs of neighboring cells within a block for appropriate normalization

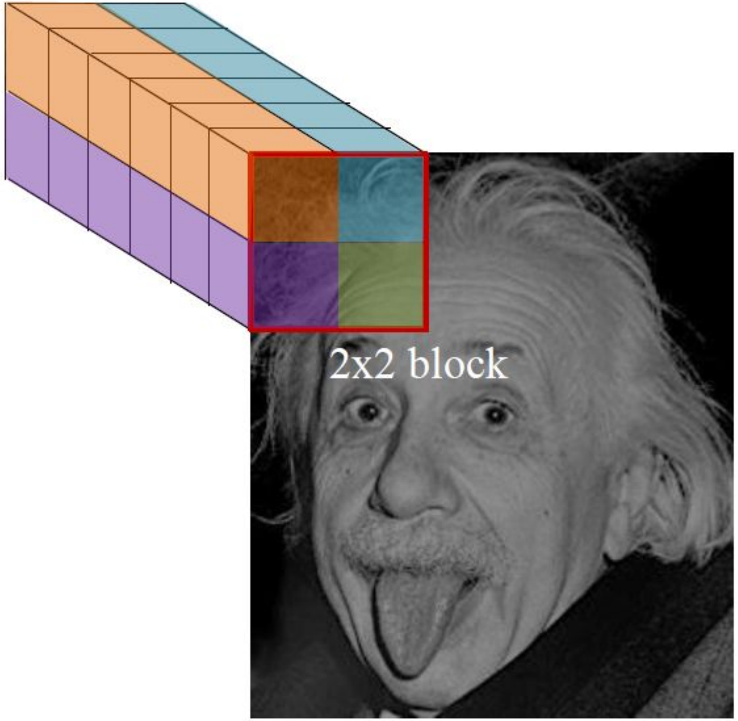




BLOCK NORMALIZATION

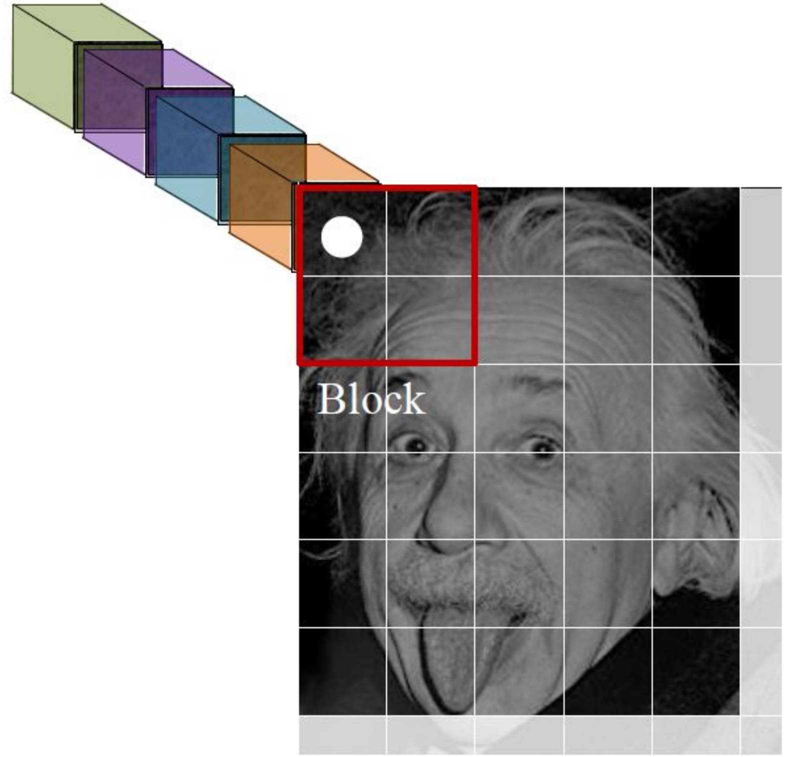
The combined histogram is 4 times
the size of cell-focused HOGs

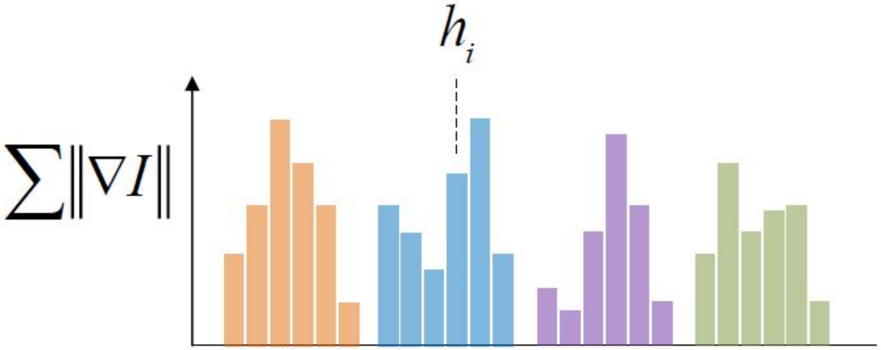




Block Normalization

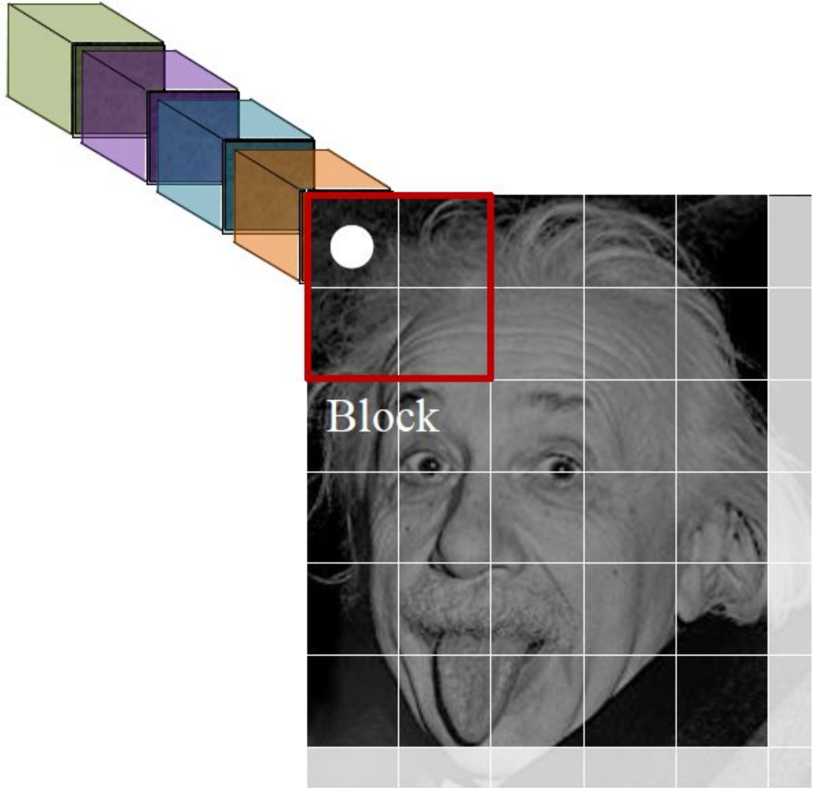
If h_i is an element in the combined histogram

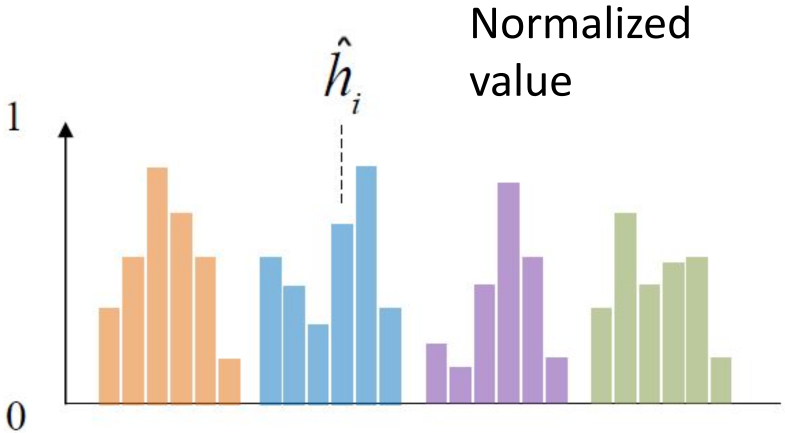




$$\hat{h}_i = \frac{h_i}{\sqrt{\sum_i h_i^2}}$$

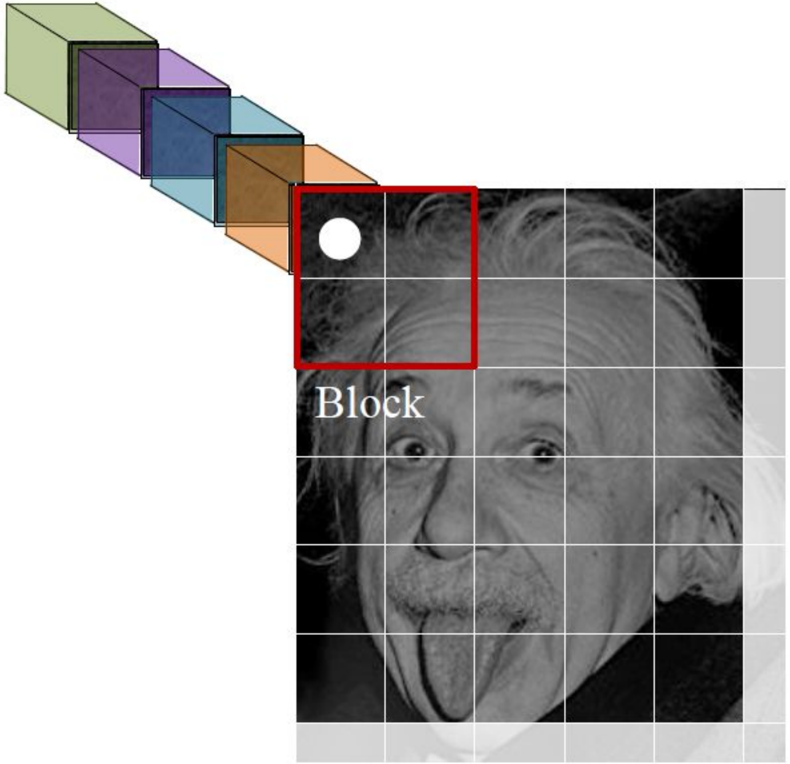
Block Normalization

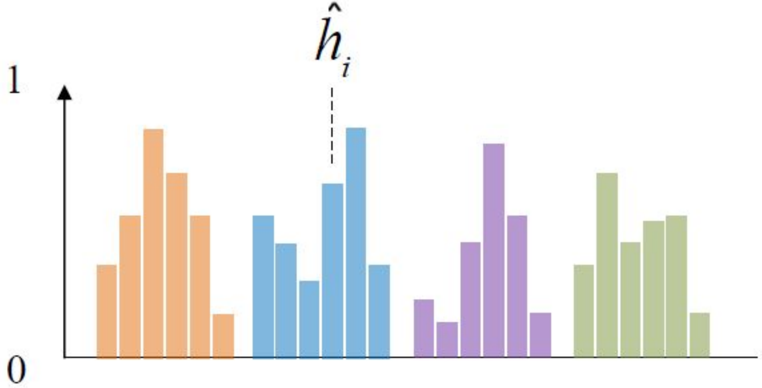




$$\hat{h}_i = \frac{h_i}{\sqrt{\sum_i h_i^2}}$$

Block Normalization

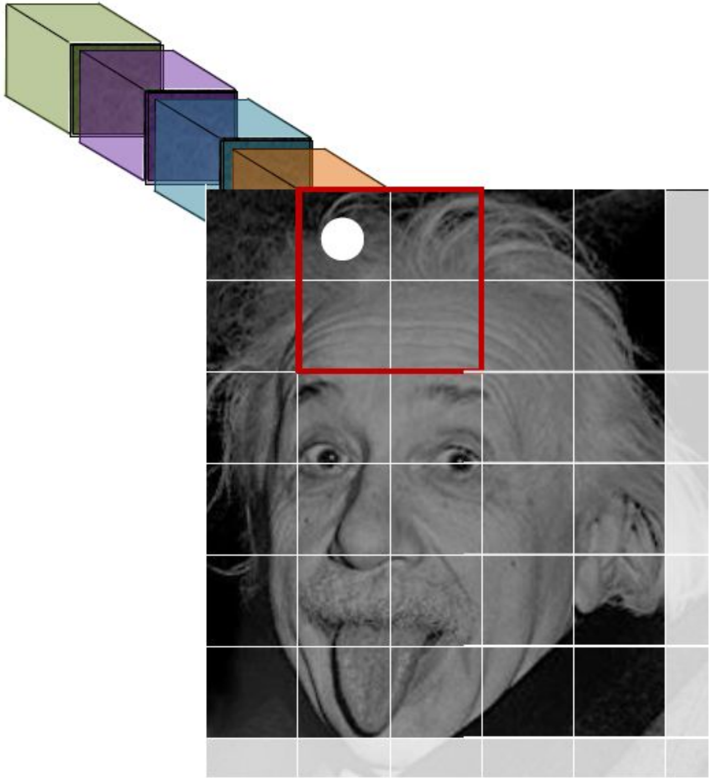




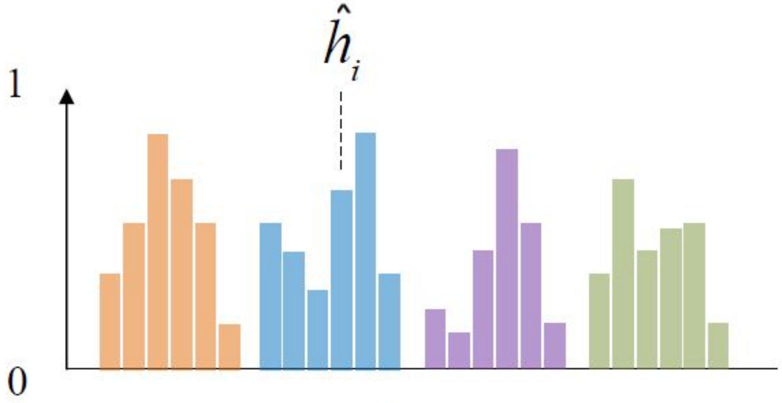
$$\hat{h}_i = \frac{h_i}{\sqrt{\sum_i h_i^2 + e^2}}$$

Preventing division by zero

Block Normalization



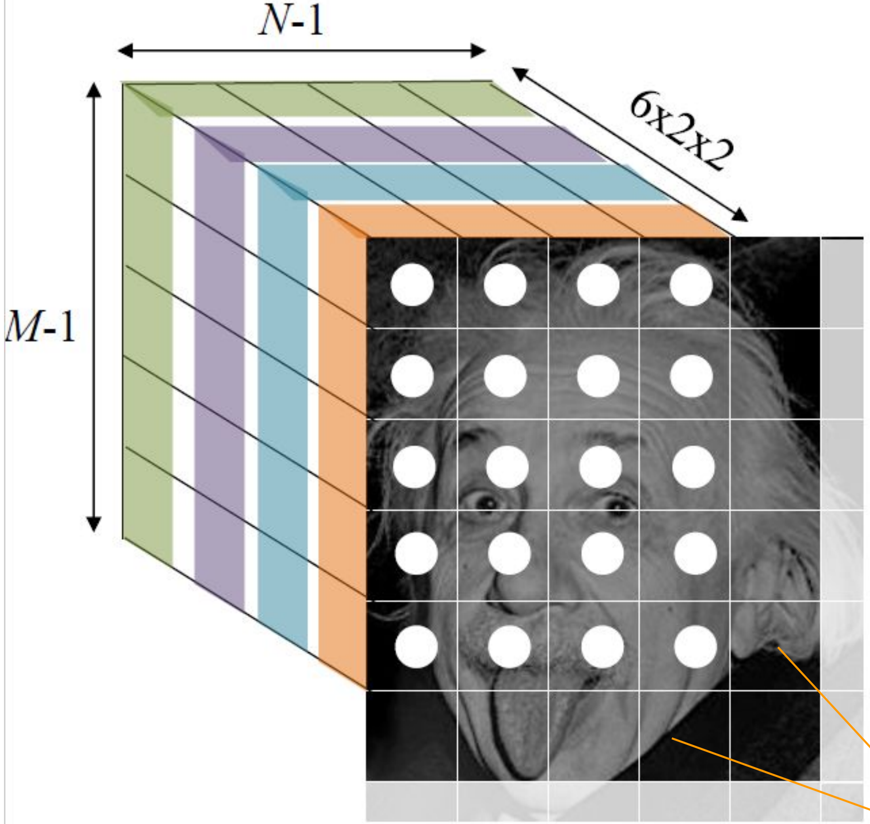
We can now do the same for the next cell using other cells in its own block

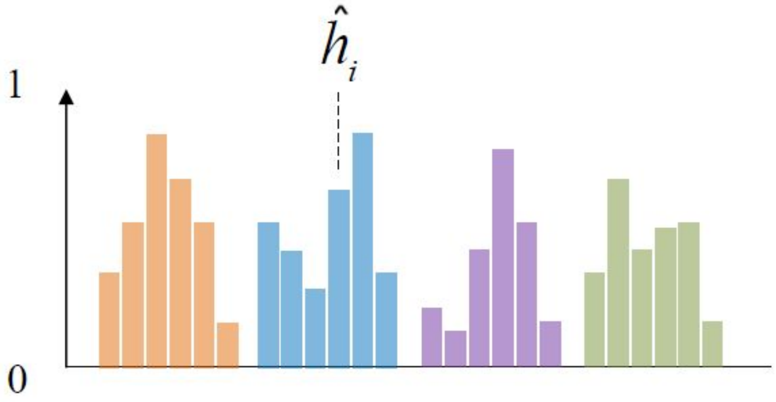


$$\hat{h}_i = \frac{h_i}{\sqrt{\sum_i h_i^2 + e^2}}$$

Preventing division by zero

Block Normalization



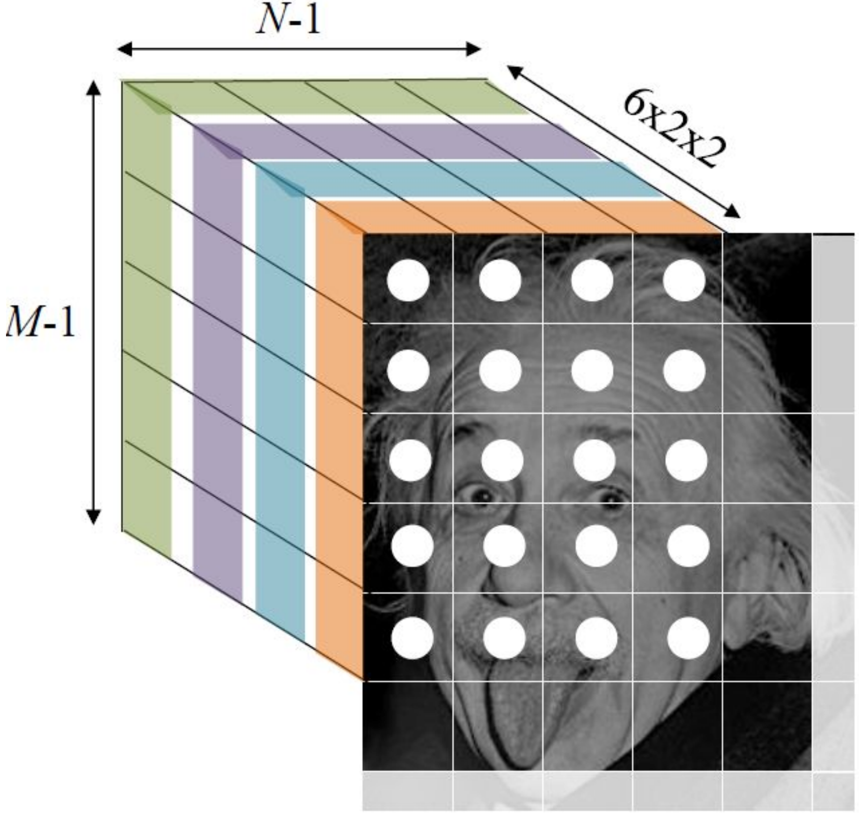


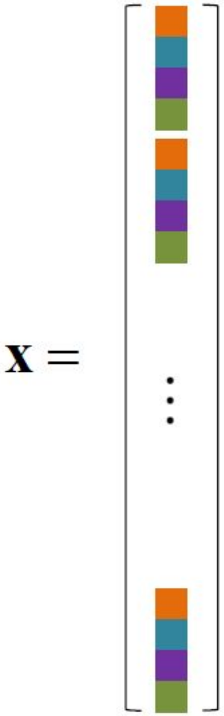
$$\hat{h}_i = \frac{h_i}{\sqrt{\sum_i h_i^2 + e^2}}$$

Preventing division by zero

However, there will be cells for which we cannot form blocks

VECTORIZATION

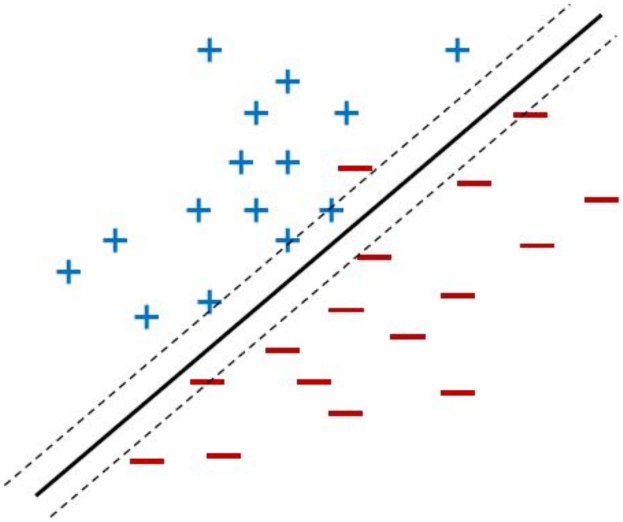




Feature descriptor of image I

Concatenating the localized feature vectors to form a global feature vector

Object Recognition with HOG



We can now use any ML algorithm (such as SVM) to build an image classifier, using HOG features, to recognize whether an image has pedestrian(s) or not.

Steps at Glance

1. Apply a vertical edge detection filter to obtain vertical edge map. Say ‘y’ is the value for a pixel in this map.
2. Apply a horizontal edge detection filter to obtain horizontal edge map. Say ‘x’ is the value for a pixel in this map.
3. Compute magnitude map using sqrt(y^2+x^2)
4. Compute angle map using the following:

$$\operatorname{atan2}(y, x) = \begin{cases} \arctan(\frac{y}{x}) & \text{if } x > 0, \\ \arctan(\frac{y}{x}) + \pi & \text{if } x < 0 \text{ and } y \geq 0, \\ \arctan(\frac{y}{x}) - \pi & \text{if } x < 0 \text{ and } y < 0, \\ +\frac{\pi}{2} & \text{if } x = 0 \text{ and } y > 0, \\ -\frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0, \\ \text{undefined} & \text{if } x = 0 \text{ and } y = 0. \end{cases}$$

The range will be -180 to 180 degrees

5. Divide the image into cells/patches (e.g. 4x4 pixels for each patch)
6. Generate a histogram over a cell that accumulates the magnitudes of the pixels in the patch using key angles as bins.

   Say, key angles are

   0, 45, 90, 135, 180, -135, -90, -45

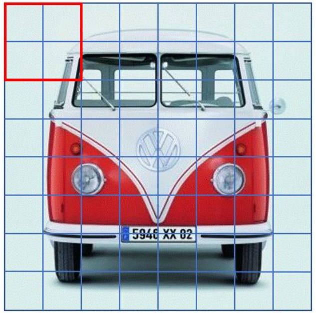
   Divide the magnitude of a pixel depending upon the proximity of the angle to the nearest key angles.

   For example,

   - If magnitude=90,angle=60, 30 goes to 90 key angle, and 60 goes to 45 key angle
   - If magnitude=90,angle=50, 10 goes to 90 key angle, and 80 goes to 45 key angle
   - If magnitude=90,angle=45, 90 goes to 45 key angle.

   Once magnitudes of all the pixels in the patch are accumulated, we have the histogram as a feature vector of the patch.

7. Group the patches into blocks. The amount of movement of the block window over the image is called stride. Concatenate the feature vectors within the block and perform normalization.



8. Concatenate the HOG features of all the blocks to generate the global feature of the given image.

Lecture Outline

Histogram of Oriented Gradients (HOG)

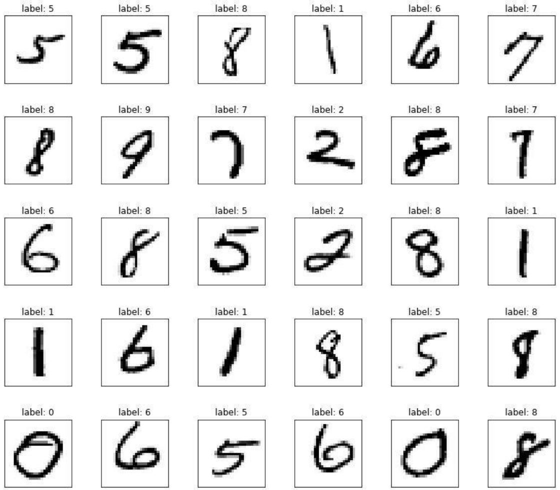
Feature Extraction (in General)

Local Binary Patterns (LBP)

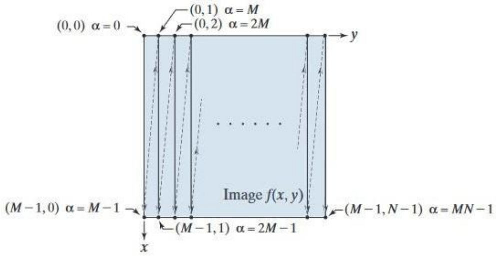
Feature Extraction

- Feature extraction is a process in machine learning and data analysis that involves identifying and extracting **relevant** features from raw data.
- Feature extraction aims to reduce data complexity (often known as “data dimensionality”) while retaining as much **relevant** information as possible.
- It yields better results than applying machine learning directly to the raw data.

An Image Dataset (MNIST)

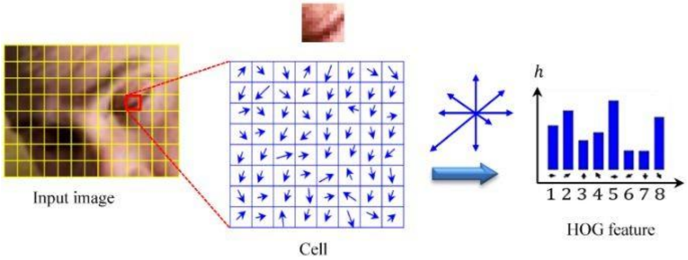


Raw Data:



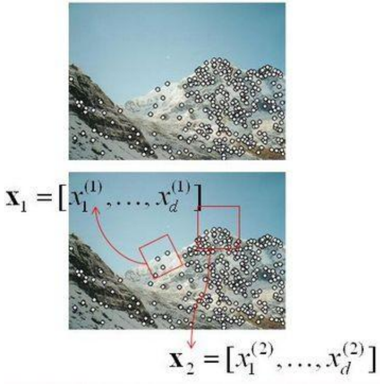
Vectorizing of image

Feature Extraction:

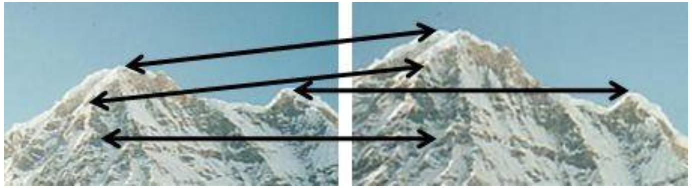


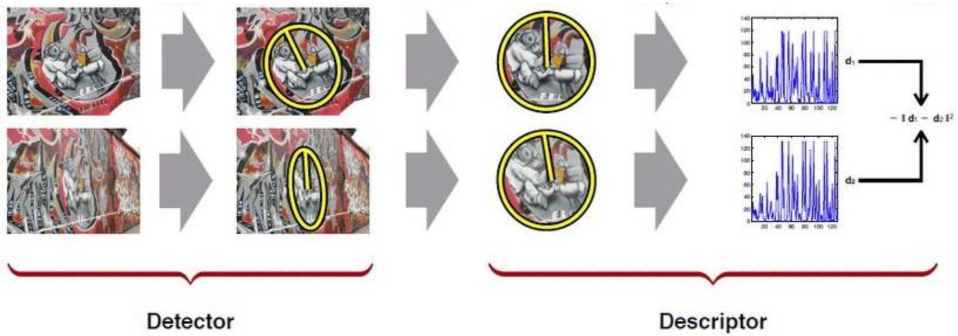
Feature Extraction in images can be divided into two steps:

- Feature detection refers to finding feature keypoints in an image.
- Feature description assigns quantitative attributes to the detected feature keypoints.



Feature Matching

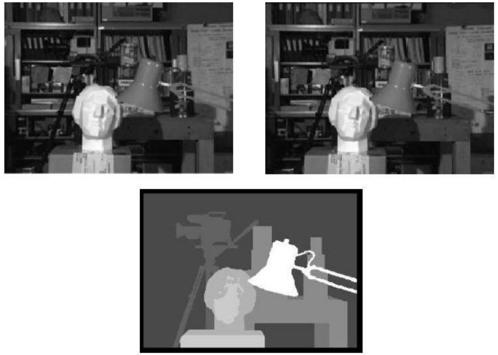




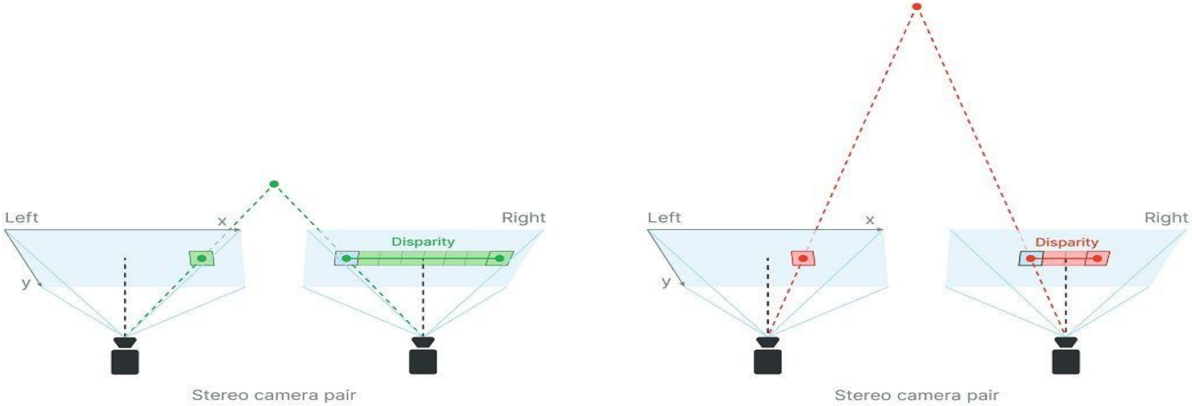
Matching

Detecting the
feature keypoints

Describing the detected feature
keypoints



Feature Matching is also useful in depth estimation



baseline

Focal length

$$\textbf{depth}\ [\text{mm}] = f_x[\text{px}] \cdot \frac{\text{baseline}\ [\text{mm}]}{\text{disparity}[\text{px}]}$$

Feature matching

- Object tracking

Matching bounding boxes across the frames

Feature matching

- Segmentation

Shapes can be distinguished based on the features

Ideal-feature

- Location-independent
- Rotation-independent
- Scale-independent
- Illumination-independent

“independent” can mean in two ways: invariant or covariant.

Invariant v/s Covariant

- A feature descriptor is invariant with respect to a set of transformations if its value remains unchanged after the application of any transformation from the family.
- A feature descriptor is covariant with respect to a set of transformations if applying any transformation from the set produces the same result in the descriptor.
- For e.g., Area of a feature:
  - Invariant, if the set of transformation is {translation, rotation, reflection}
  - Covariant, if the set of transformation is {scaling}

Local vs Global Features

Local features are specific to small regions or patches of an image. They capture details that are relevant within a limited context. They are useful for tasks requiring fine-grained analysis. E.g. keypoints, corners, edges, etc.

Global features describe the overall characteristics of an image, considering the entire image. They are useful for high-level analysis and classification tasks. E.g. histogram, aspect ratio, etc.

Lecture Outline

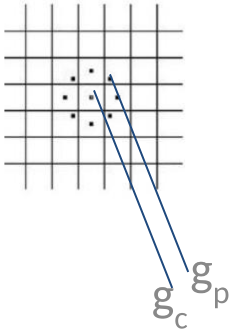
Histogram of Oriented Gradients (HOG)

Feature Extraction (in General)

Local Binary Patterns (LBP)

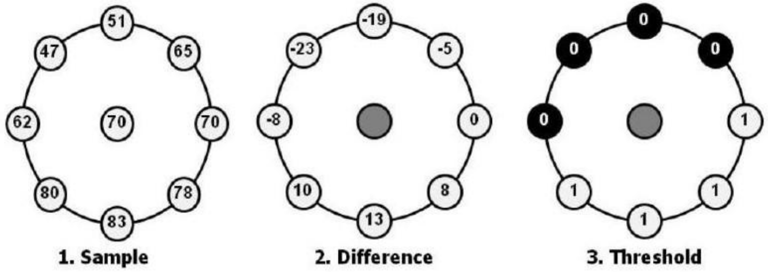
LBP

- Local Binary Pattern
  - texture and local pattern detection
  - textures have no specific definition
    - complex patterns having more sub-patterns



The value of the LBP code of a pixel (x_c, y_c) is given by:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \qquad s(x) = \begin{cases} 1, if\ x \geq 0; \\ 0, otherwise. \end{cases}$$

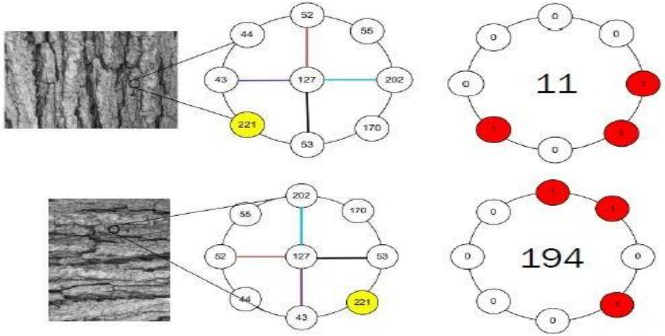


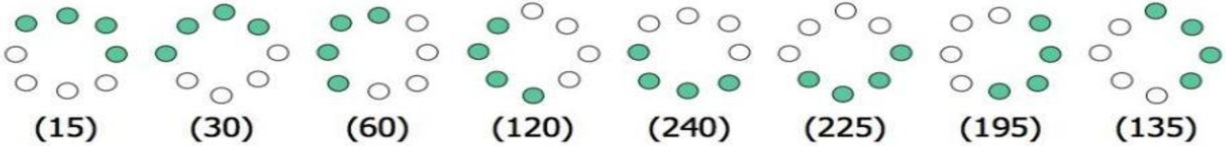
1*1 + 1*2 + 1*4 + 1*8 + 0*16 + 0*32 + 0*64 + 0*128 = 15

LBP

- Invariant to
  - ilumination
    - shadow, reflection, brightness
    - relative difference between intensities remain same
  - rotation?

No



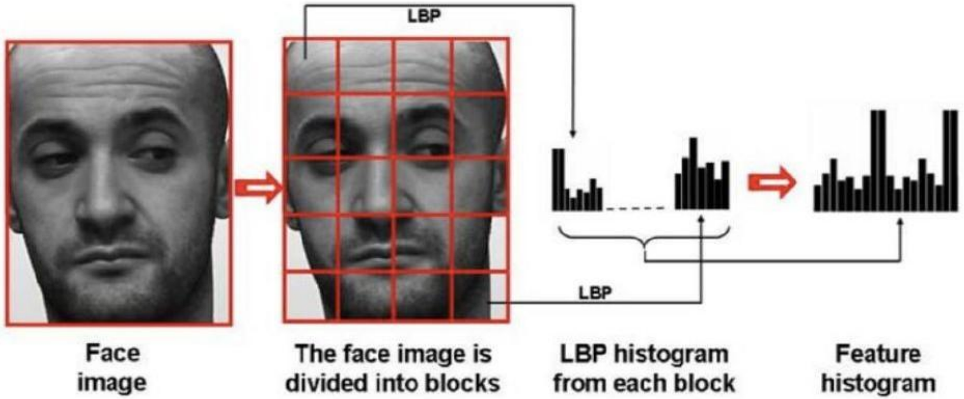


LBP

- Local Binary Pattern
  - 8 neighborhood gives 256 possible LBP codes
    - each pixel gets one of the codes
  - LBP histogram 256D
    - probability of occurrence of each LBP code

LBP

- LBP to global descriptor



Concatenation of local LBP histograms