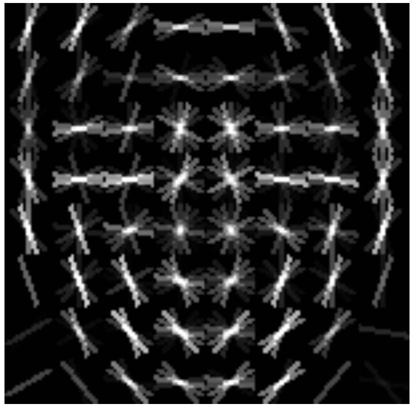
IMAGE DESCRIPTORS: HISTOGRAM OF GRADIENTS (HOG)

Artificial Vision



Class 5: HOG

Today's plan

- What are and why we need image descriptors?
- A particular kind of image descriptor (HOG)
- Examples of practical use of HOG

Problem 1: Image Classification

Given an image, how can we automatically label it choosing between the labels 'building' and 'nature'?

Which **visual characteristics** of the image can be used for this goal?

Problem 1: Image Classification

Given an image, how can we automatically label it choosing between the labels 'building' and 'nature'?

Which **visual characteristics** of the image can be used for this goal?

Problem 2: Image Retrieval

Given a 'building' image (query), how to retrieve other 'building' images in a database?

Why we need a descriptor?

To solve real world problems (image classification, image retrieval), we need to find a connection between:

- a matrix of pixels (raw representation),
- what humans see in an image (face, smile, emoticon).

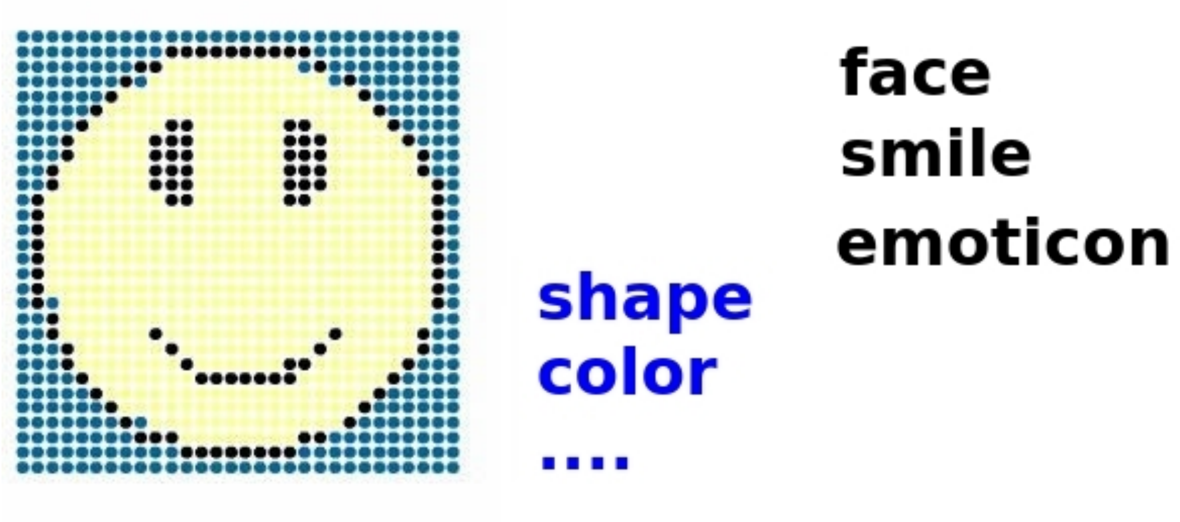


Image descriptors allow to describe and represent the image by quantities (color, shape, regions, textures and motion) closer to the visual characteristics perceived by humans.

Algorithm for image retrieval

Given a 'building' image (query), how to retrieve other 'building' images in a database?

Query image

Database

1. Define the image descriptor
2. Extract the image descriptors of the database images

3. Given a query image, extract its descriptor
4. Sort the database images according to the similarity with the query image.

Algorithm for image classification

Labeled images (training set)

Building **Building** **Nature** **Nature**

Test image

'Building' or 'Nature'?

Training:

1. Define image descriptors
2. Use training set to extract their descriptors
3. Train a model

Test:

4. Given a test example extract its descriptor
5. Apply the model and compare with the training examples to decide its label

General approach to classification problems

Let's suppose for now that the descriptor is simply the mean color....

Labeled images (training set)

Building

Building

Nature

Nature

Test image

'Building' or 'Nature'?

Training phase:

- Represent each image of the training set by its descriptor
- Store the descriptors and class labels of the training samples (labeled images)

R	G	B	Label
20	30	200	1
34	166	111	1
12	220	222	0
25	244	30	0

- **Model**: if G>200, -> label 1. Otherwise, label 0.

General approach to classification problems

Let's suppose for now that the descriptor is simply the mean color....

Labeled images (training set)

Building

Building

Nature

Nature

Test image

'Building' or 'Nature'?

Test phase:

- Compute the descriptor of the test image
- Apply the model to compare the descriptor of the test image to the descriptors of the training images in order to determine its class.

R	G	B	Label
20	30	200	1
34	166	111	1
12	220	222	0
25	244	30	0
233	**55**	**211**	**?**

General approach to classification problems

Training phase:

- Represent each image of the training set by its descriptor
- Store the descriptors and class labels of the training samples (labeled images)

How to choose the descriptor?

The **descriptor (or feature vector)** should describe the image in a way that is invariant to all the image changes that are suitable to our application (e.g. color, illumination, noise etc.)



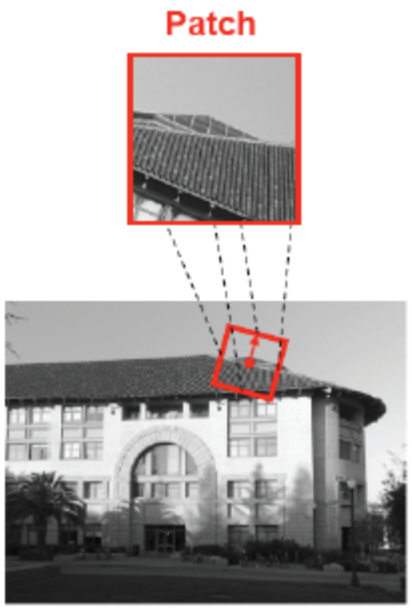


What do have in common all these buildings?

What does distinguish them from images of natural lands?

How to choose the descriptor?

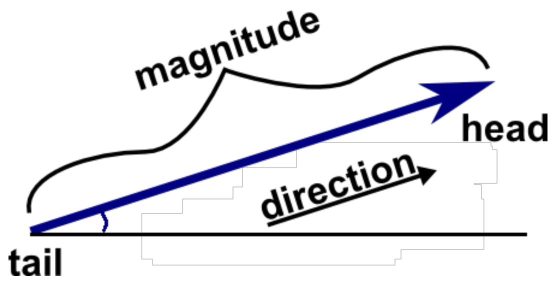
Can we discriminate objects based on their local shape and appearance?



Is the gradient structure characteristic of local or global shape and appearance?

Histogram of gradient (HOG)

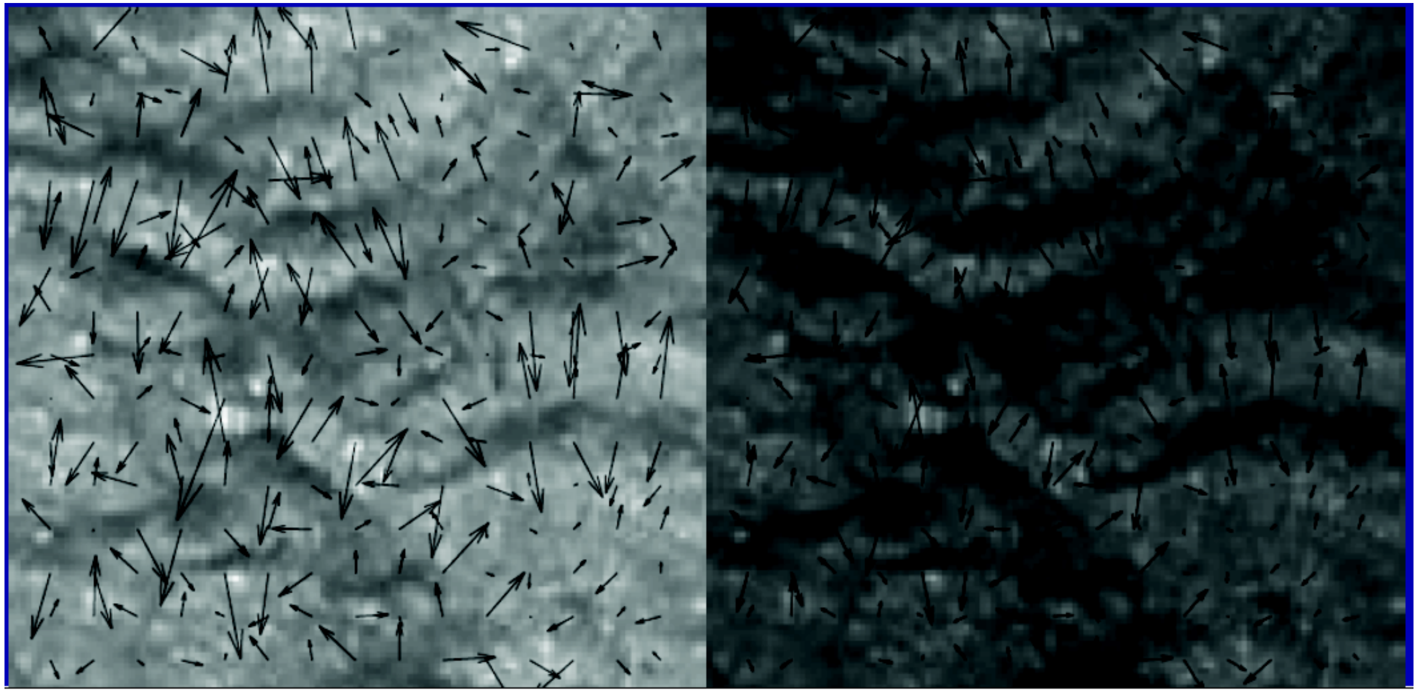
Remember what is the image gradient....



- The image gradient at each pixel is a vector.
- As a vector, it has a magnitude and a direction.

Histogram of gradient (HOG)

Would the gradient magnitude be useful?

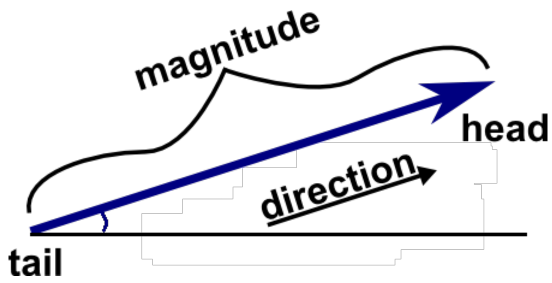


Gradient magnitude is affected by illumination changes!

But the direction isn't!

Histogram of gradient (HOG)

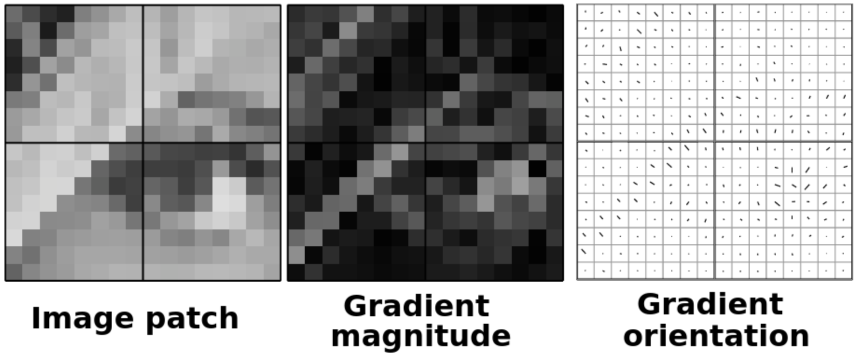
What could be the histogram of gradient



...knowing that the gradient is characterized by the two quantities?

Histogram of gradient (HOG)

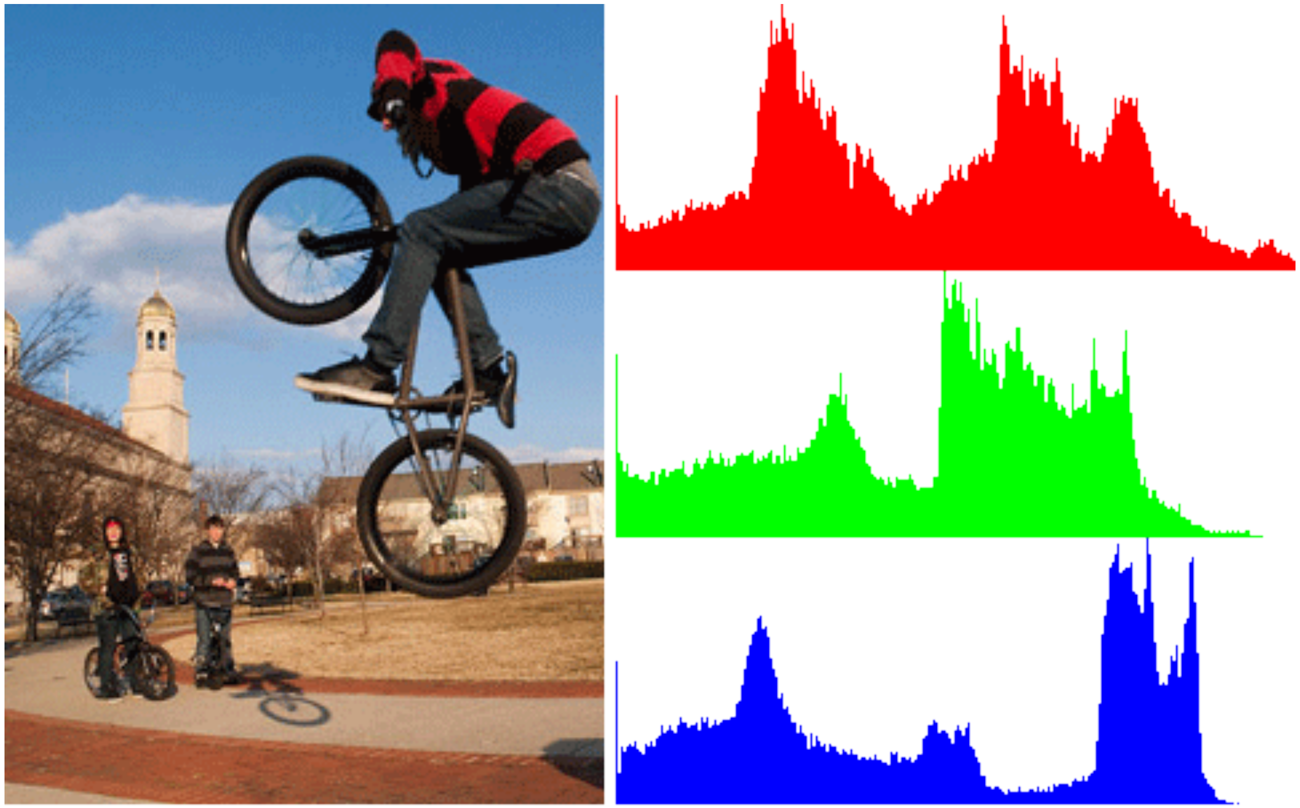
Gradient of an image patch



How to obtain the overall orientation of the pixels gradient?

Histogram of gradient (HOG)

Remember what is the histogram of color...

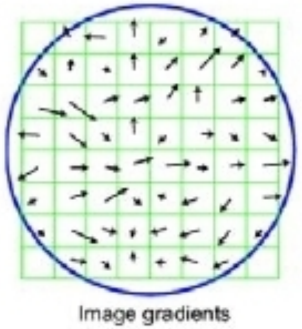


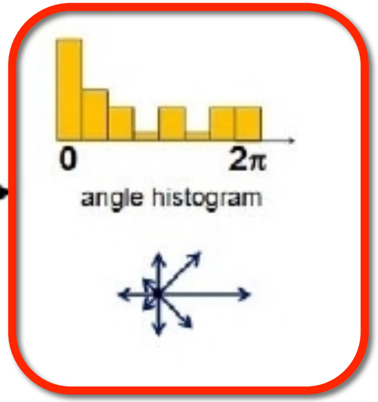
Histogram of gradient (HOG)

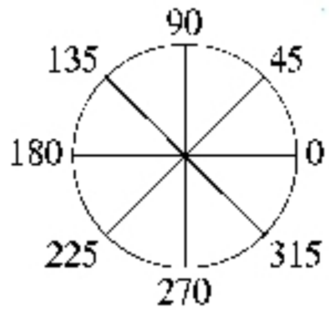
Histogram of gradient orientations

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$



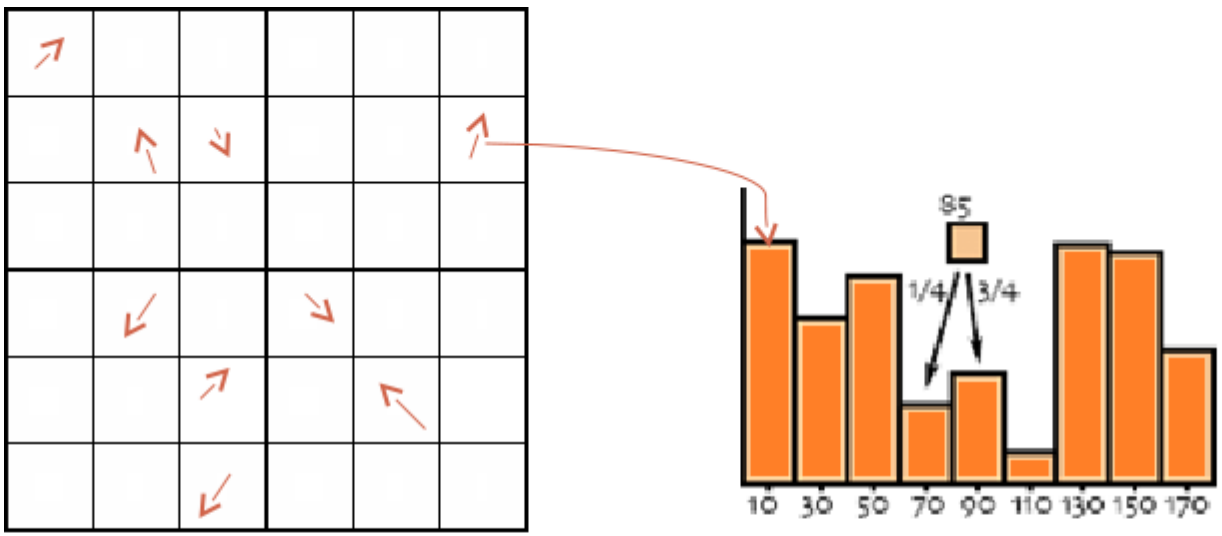




- The gradient orientation is an angle
- Count occurrences of gradient orientation in a patch
- Quantize to 8 bins, each bins cover 45 degrees
- Visual representation of the histogram

Histogram of gradient (HOG)

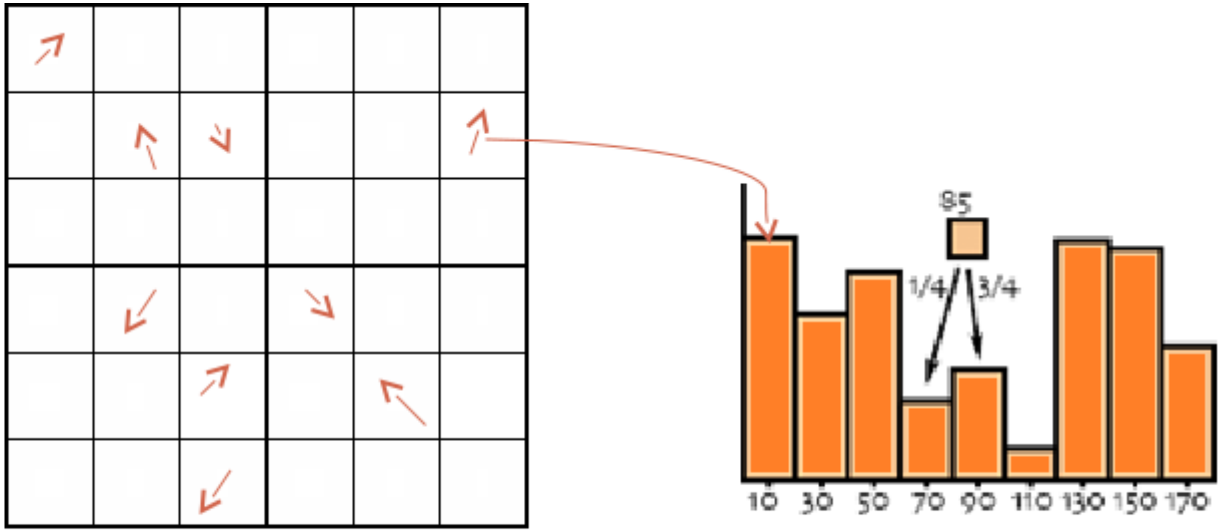
Histogram of gradient orientations



- From 0 to 180 degrees, 9 bins, 20 degrees per bin
- θ= 85 degrees
- To which bin it contributes?

Histogram of gradient (HOG)

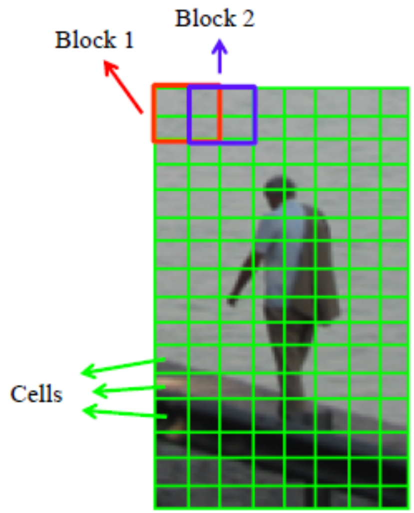
Histogram of gradient orientations



- Compute the distance to adjacent bin centers (from Bin 70 and Bin 90 are 15 and 5 degrees, respectively).
- Divide the distance by the size of the bins (distance/binsize): 5/20=1/4, 15/20=3/4
- Weight the contribution by the gradient magnitude

Histogram of gradient (HOG)

- Divide the image into small connected regions called **cells**.
- Compute a local histogram for each cell weighted by gradient magnitude
- Simply concatenate the histogram of the cells.



Contrast - normalization

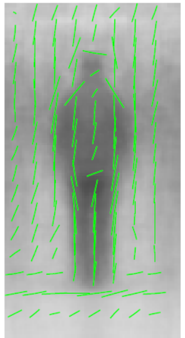
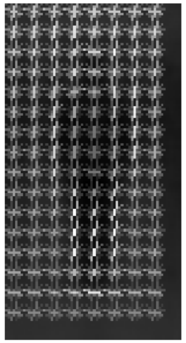
- Gradient strengths vary over a wide range owing to local variations in illumination and foreground-background contrast.

- **How to achieve invariance to changes in illumination or shadowing?**

- Compute a measure of intensity across a larger region than a cell (a block)

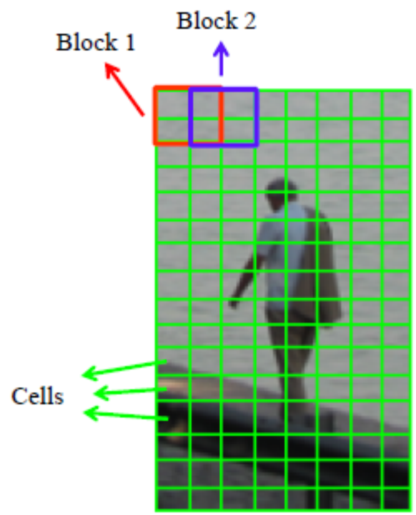
- Normalize all cells within the block with this intensity value

- L_2 normalization: $L_2 = \sqrt{(||v||^2_2 + \varepsilon^2)}$, ε is a regularization parameter.

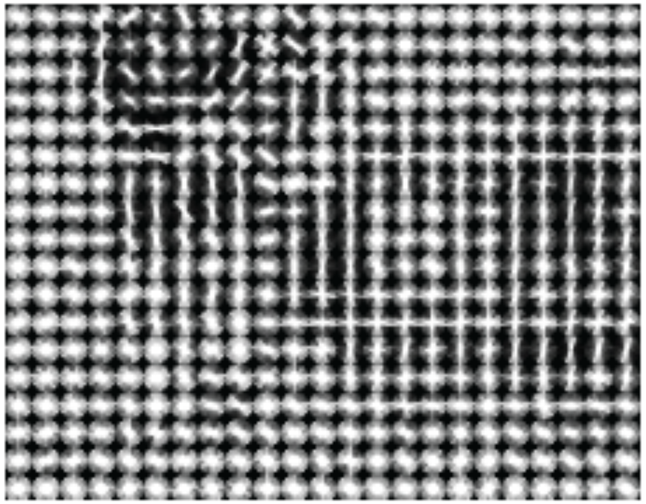
Histogram of gradient (HOG)

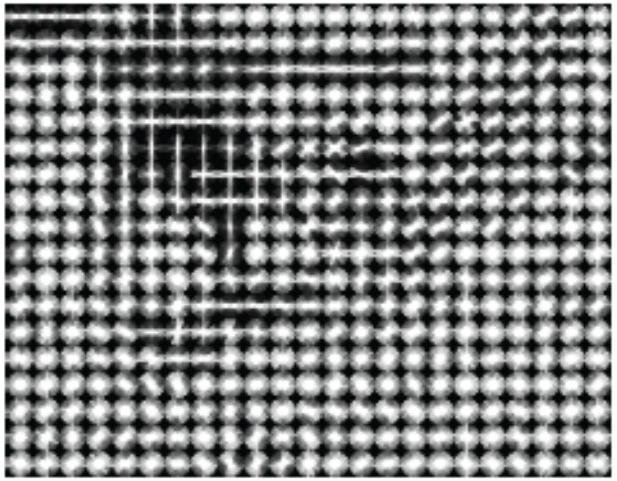
- For a 64x128 image
- Divide the image in cells of 8x8 pixels (8x16 cells)
- Group cells into blocks of 2x2 cells (16x16 pixels) of 50% overlap
- Total number of blocks: 7x15= 105
- Quantize the gradient orientation into 9 bins
- Concatenate histograms: 105 x 4 x 9 = 3780 feature vector



Histogram of gradient (HOG)

Can we say that the HOG is able to describe local shape and appearance?





Does the HOG descriptor carry information about the gradient or edge positions?

Compute gradient in practice

- Convolve the image with discrete derivative mask:
  - $D_x=[-1, 0, 1]$, $D_y=[1, 0, -1]^T$
- Angles: $\tan^{-1}(D_y/D_x)$
- Magnitude: $\sqrt{(D_y^2 + D_x^2)}$

-1	0	1

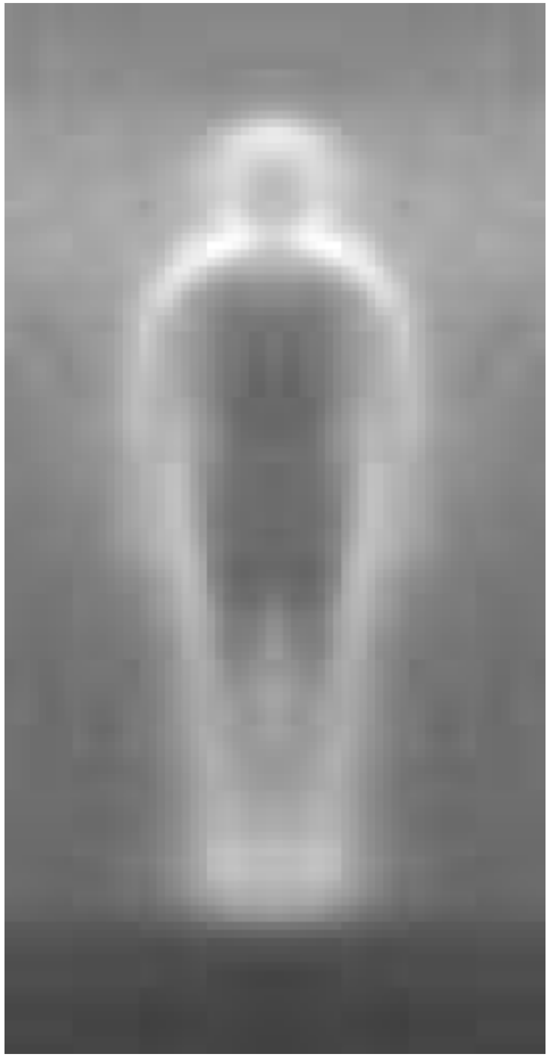
centered

-1	1

uncentered

1	-8	0	8	-1

cubic-corrected



0	1
-1	0

diagonal

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

Sobel

Compute gradient with MATLAB

```
> Im = double(im);

> Dx = [-1,0,1];

> Dy = -Dx';

> gradx = imfilter(double(Im),Dx);
> grady = imfilter(double(Im),Dy);
> angles=atan2(grady,gradx);
> magnit=sqrt(((gradx.*gradx)+
(grady.*grady)));
```

Once we have the descriptor, how do we classify the images?

Image classification

Given an image, assign it a label ('building' or 'nature')

K-Nearest Neighbors for classification

- The feature vector of the image is a point in our feature space
- The image is classified by assigning the label which is most frequent among the k training samples nearest to the test point.
  - k is a user-defined constant (How to choose k?)
  - Can the classification change for different k?

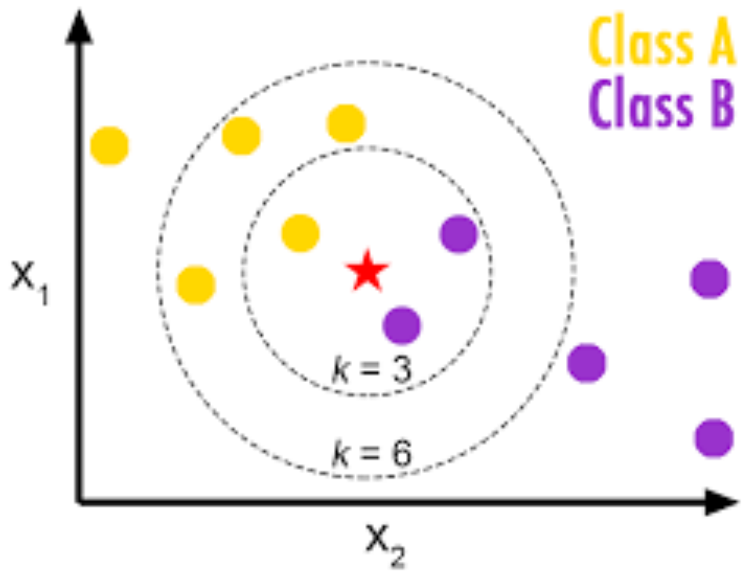
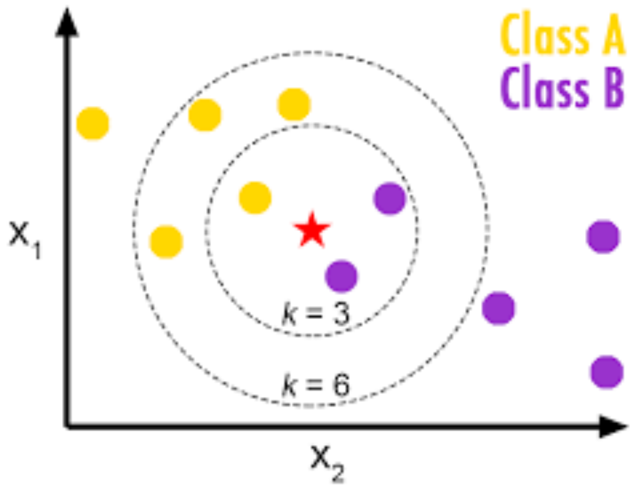


Image retrieval

Given an image (query), find all similar images in the database.

K-Nearest Neighbors for retrieval

- The query is an unlabeled vector in our feature space
- Retrieve the k-closest neighbors as the relevant items to a query
  - k is a user-defined constant
  - Database images do not necessarily have labels.



Pedestrian detection

Pedestrian detection

Why is the problem difficult?

- Wide variety of articulated poses
- Variable appearance/clothing
- Complex backgrounds
- Unconstrained illumination
- Occlusions
- Different scales




Pedestrian detection

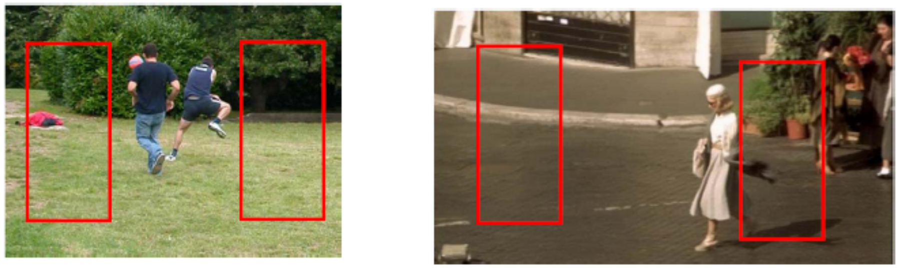
- Trasform the detection problem into a binary classification problem:

  "Is a given window representing a pedestrian or not?"

- We need labeled examples (training data)!!

  - Positive data – 1208 positive window examples

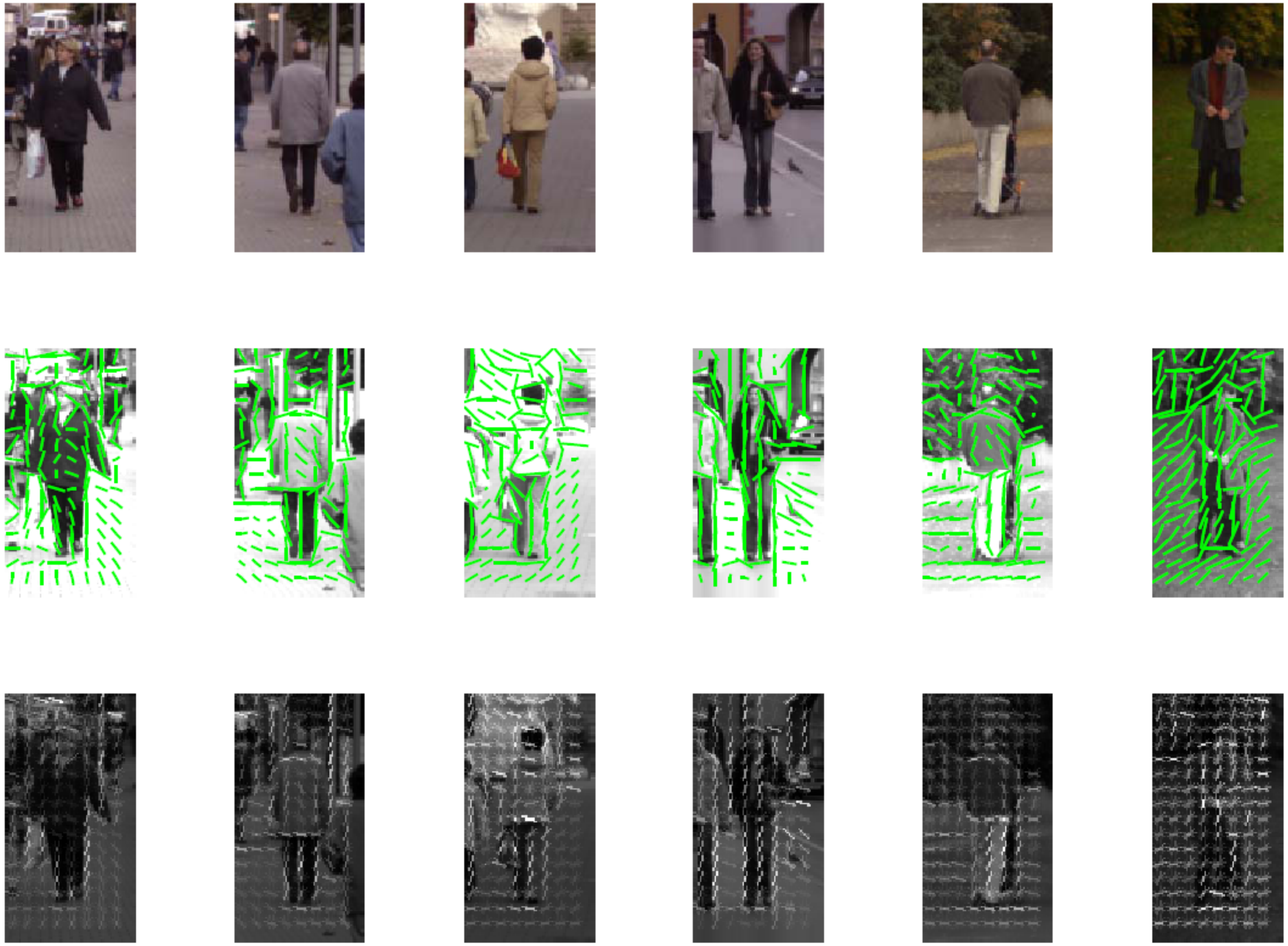
  - Negative data – 1218 negative window examples (initially)



Dalal and Triggs, CVPR 2005

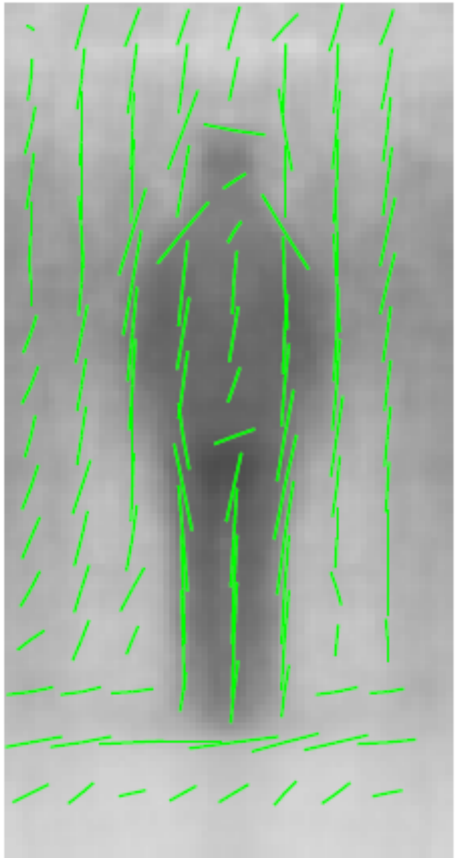
Pedestrian detection

Compute HOG descriptor for all training samples

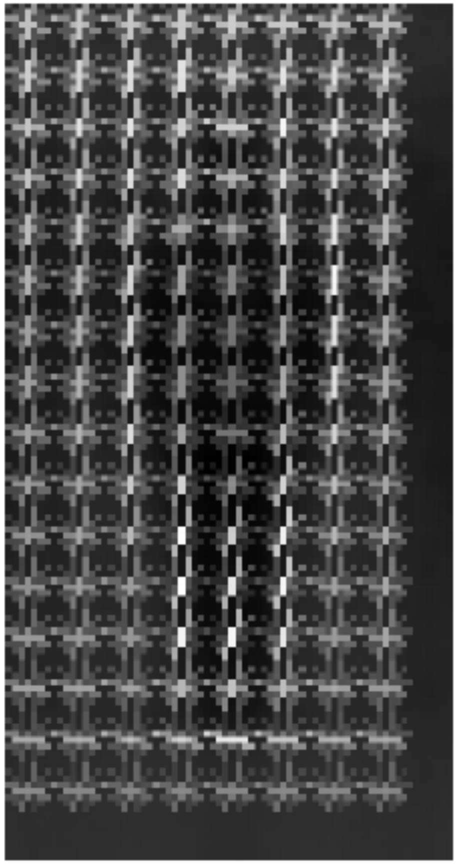


Pedestrian detection

Averaged positive examples



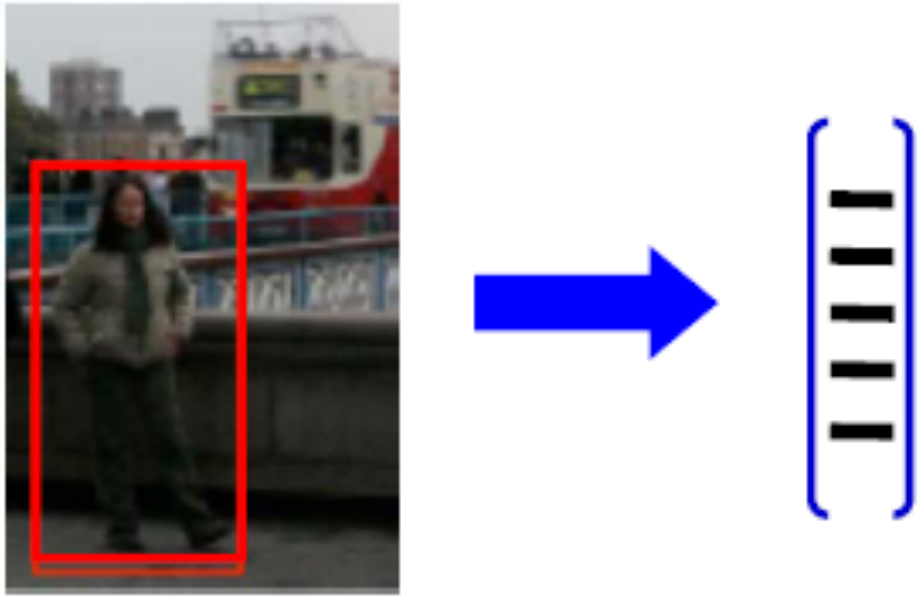
Predominant direction



Histograms of gradients

Training phase

Compute HOG for each training example and train a classifier.



$$\mathbf{x}_i \in \mathbb{R}^d$$

What classifier to use?

- knn
- There are other classifiers too (SVM), etc.

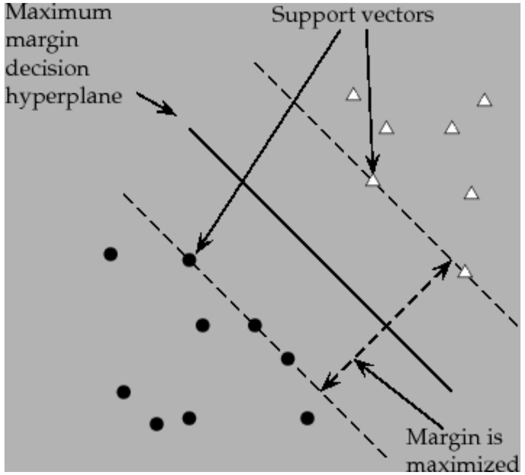
Intuition behind Linear SVM

Project the data in a feature space ($w^T x$), where w is a matrix and find a decision boundary between two classes that is maximally far from any point in the training data:

Formally, find matrix w such that the function:

$$f(x) = w^T x + b$$

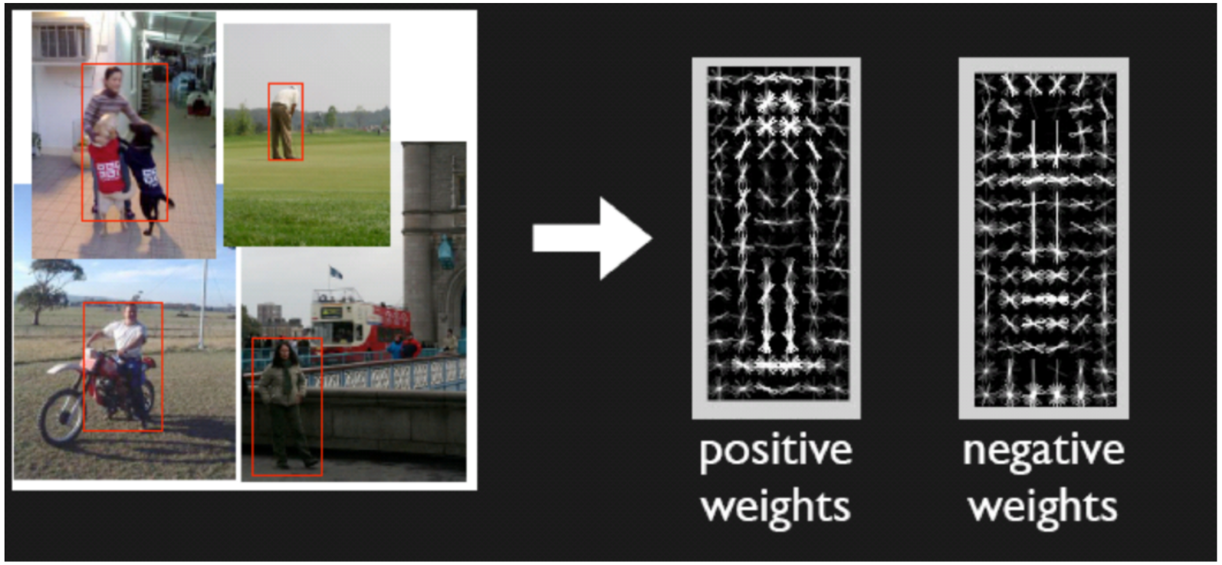
separates the two categories with the maximum margin (optimization problem).



The subset of the training data which defines the position of the separator are the support vectors.

Learnt model

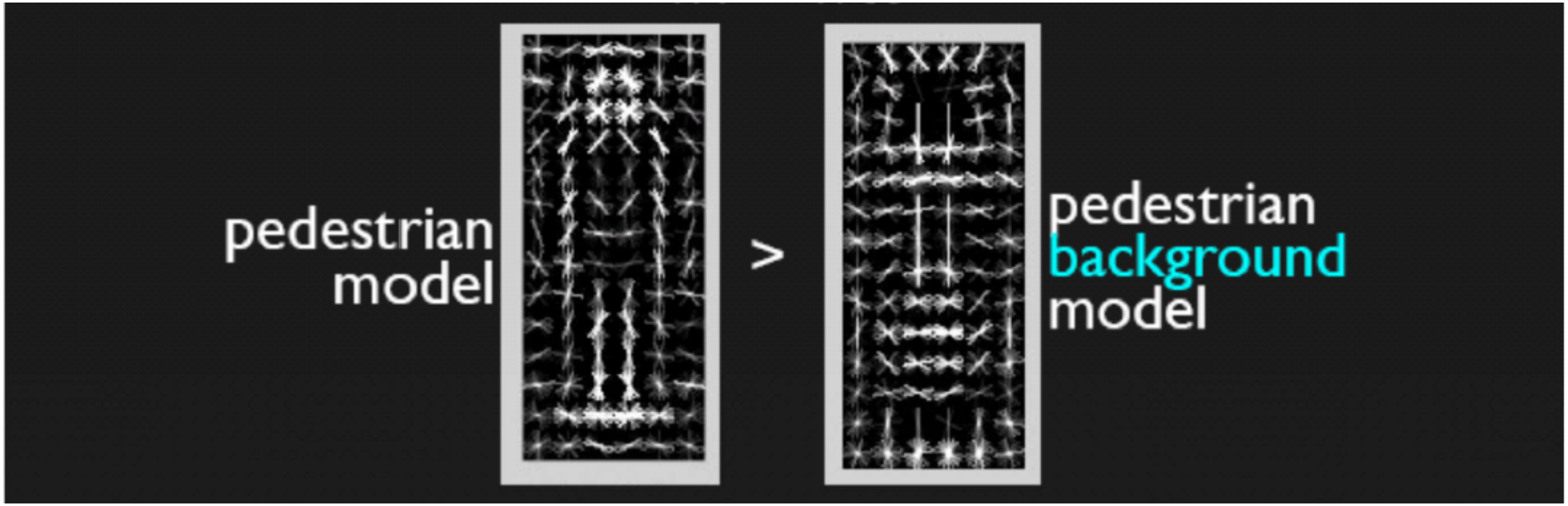
When we train a SVM, we get a weight vector w



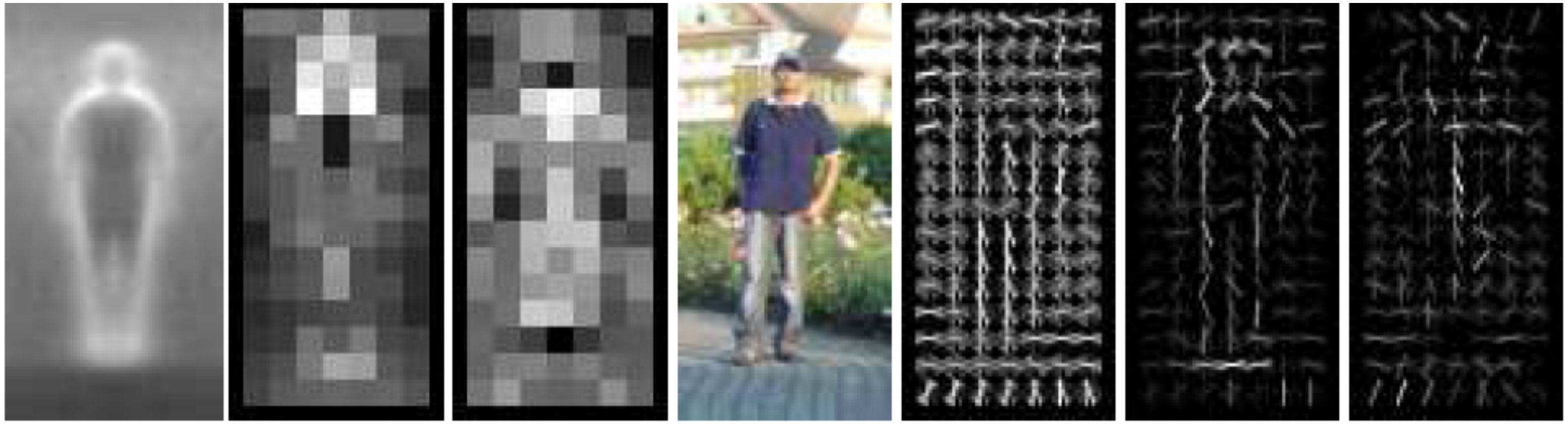
Slide from Deva Ramanan

- Positive weights show edge orientations that should be present in images of pedestrians.
- Negative weights show edge orientations that should not be present in an image region containing a pedestrian (horizontal edges in the region of the legs)

Meaning of negative weights



Test phase (detection)



(a) (b) (c) (d) (e) (f) (g)

Training:
(a) Get the average gradient image over training examples,
(b) extract HOGs and train a classifier (SVM) so that we get max positive SVM weight in the block centered on that pixels,
(c) same as (b) for negative SVM weights.

Test:
(d) Given a test image, classify every image window/region (using sliding window) by
(e) extract its HOG descriptor
(f) HOG descriptor multiplied by positive SVM weights
(g) HOG descriptor multiplied by negative SVM weights,

Compute which image gets the higher response -> Pedestrian vs. No-pedestrian.

Pedestrian detection results

Common pipeline

- Extract features which are discriminant for your problem
  - each image is represented by a high-dimensional feature vector.
- Use a machine learning algorithm
  - this always requires to first extract features from all images of the training set.
- Make predictions (classification) on the test images to detect the looked for model.
  - K-nn
  - Support vector machine classifier

Test

- Image descriptor is…
- Classification consists of..
  - Does the classification need a label?
- Retrieval consists of…
  - Does the retrieval need a label?
- Histograms of gradients is based on orientation instead of magnitude in order to ….
- The K-nn classifier is based on...
  - The higher k is applied, ...
- The SVM classifier is based on...
  - Support vectors are..
  - SVM projects data into
- Pedestrian detection is based on...
  - The difficulty comes from...
  - We need positive and negative traaining examples in order to...