# Computer Vision - Lecture 8

## Sliding-Window based Object Detection

**21.11.2016**

Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de

leibe@vision.rwth-aachen.de



## Course Outline

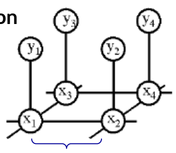
- Image Processing Basics
- Segmentation
  - Segmentation and Grouping
  - Segmentation as Energy Minimization
- Recognition & Categorization
  - Sliding-Window Object Detection
  - Image Classification
- Local Features & Matching
- 3D Reconstruction
- Motion and Tracking



## Recap: MRFs for Image Segmentation

- MRF formulation

Unary potentials $\phi(x_i, y_i)$




Pairwise potentials $\psi(x_i, x_j)$

⇒ Minimize the energy

$$E(\mathbf{x},\mathbf{y}) = \sum_i \phi(x_i, y_i) + \sum_{i,j} \psi(x_i, x_j)$$

Data (D) | Unary likelihood | Pair-wise Terms | MAP Solution

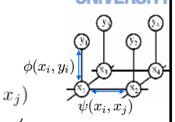


Slide adapted from Phil Torr

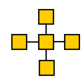
## Recap: Energy Formulation

- Energy function

$$E(\mathbf{x},\mathbf{y}) = \underbrace{\sum_i \phi(x_i, y_i)}_{\text{Unary potentials}} + \underbrace{\sum_{i,j} \psi(x_i, x_j)}_{\text{Pairwise potentials}}$$




- Unary potentials $\phi$
  - Encode local information about the given pixel/patch
  - How likely is a pixel/patch to belong to a certain class (e.g. foreground/background)?
- Pairwise potentials $\psi$
  - Encode neighborhood information
  - How different is a pixel/patch's label from that of its neighbor? (e.g. based on intensity/color/texture difference, edges)



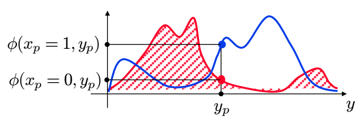


## Recap: How to Set the Potentials?

- Unary potentials
  - E.g. color model, modeled with a Mixture of Gaussians

$$\phi(x_i, y_i; \theta_\phi) = \log \sum_k \theta_\phi(x_i, k) p(k|x_i) \mathcal{N}(y_i; \bar{y}_k, \Sigma_k)$$

⇒ Learn color distributions for each label






## Recap: How to Set the Potentials?

- Pairwise potentials
  - Potts Model

$$\psi(x_i, x_j; \theta_\psi) = \theta_\psi \delta(x_i \neq x_j)$$

    - Simplest discontinuity preserving model.
    - Discontinuities between any pair of labels are penalized equally.
    - Useful when labels are unordered or number of labels is small.
  - Extension: "Contrast sensitive Potts model"

$$\psi(x_i, x_j, g_{ij}(\mathbf{y}); \theta_\psi) = -\theta_\psi g_{ij}(\mathbf{y}) \delta(x_i \neq x_j)$$

where

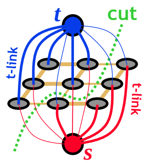
$$g_{ij}(\mathbf{y}) = e^{-\beta \|y_i - y_j\|^2} \qquad \beta = \frac{1}{2}\left(\text{avg}\left(\|y_i - y_j\|^2\right)\right)^{-1}$$

⇒ Discourages label changes except in places where there is also a large change in the observations.

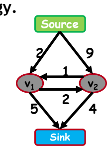
## Recap: Graph-Cuts Energy Minimization

- Solve an equivalent graph cut problem
  1. Introduce extra nodes: source and sink
  2. Weight connections to source/sink (t-links) by $\phi(x_i = s)$ and $\phi(x_i = t)$, respectively.
  3. Weight connections between nodes (n-links) by $\psi(x_i, x_j)$.
  4. Find the minimum cost cut that separates source from sink.

  ⇒ Solution is equivalent to minimum of the energy.




- s-t Mincut can be solved efficiently
  - Dual to the well-known max flow problem
  - Very efficient algorithms available for regular grid graphs (1-2 MPixels/s)
  - Globally optimal result for 2-class problems






## Recap: When Can s-t Graph Cuts Be Applied?

Unary potentials / Pairwise potentials

$$E(L) = \sum_p E_p(L_p) + \sum_{pq \in N} E(L_p, L_q) \qquad L_p \in \{s,t\}$$

t-links / n-links

- s-t graph cuts can only globally minimize binary energies that are submodular. [Boros & Hummer, 2002, Kolmogorov & Zabih, 2004]

*E(L)* can be minimized by *s-t* graph cuts $\Leftrightarrow$ $\boxed{E(s,s)+E(t,t) \leq E(s,t)+E(t,s)}$

Submodularity ("convexity")

- Submodularity is the discrete equivalent to convexity.
  - Implies that every local energy minimum is a global minimum.

  ⇒ Solution will be globally optimal.



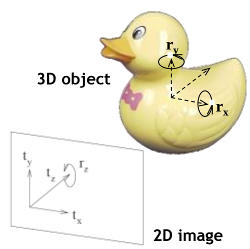
## Topics of This Lecture

- Object Recognition and Categorization
  - Problem Definitions
  - Challenges
- Sliding-Window based Object Detection
  - Detection via Classification
  - Global Representations
  - Classifier Construction
- Classification with SVMs
  - Support Vector Machines
  - HOG Detector
- Classification with Boosting
  - AdaBoost
  - Viola-Jones Face Detection



## Object Recognition: Challenges

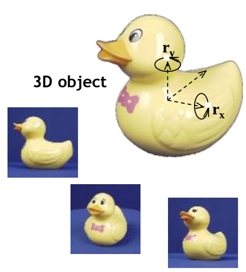
- Viewpoint changes
  - Translation
  - Image-plane rotation
  - Scale changes
  - Out-of-plane rotation
- Illumination
- Noise
- Clutter
- Occlusion






## Appearance-Based Recognition

- Basic assumption
  - Objects can be represented by a set of images ("appearances").
  - For recognition, it is sufficient to just compare the 2D appearances.
  - No 3D model is needed.

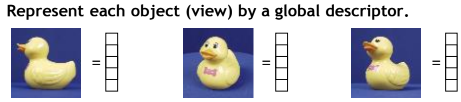



⇒ Fundamental paradigm shift in the 90's



## Global Representation

- Idea
  - Represent each object (view) by a global descriptor.

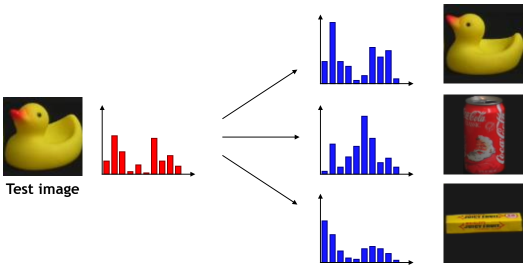



  - For recognizing objects, just match the descriptors.
  - Some modes of variation are built into the descriptor, the others have to be incorporated in the training data.
    - E.g., a descriptor can be made invariant to image-plane rotations.
    - Other variations:

| | |
|---|---|
| Viewpoint changes | Illumination |
| – Translation | Noise |
| – Scale changes | Clutter |
| – Out-of-plane rotation | Occlusion |

## Appearance based Recognition

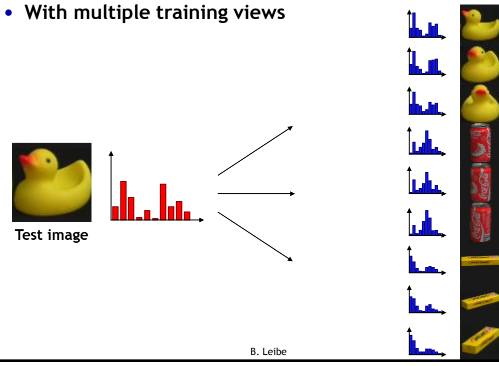
- Recognition as feature vector matching





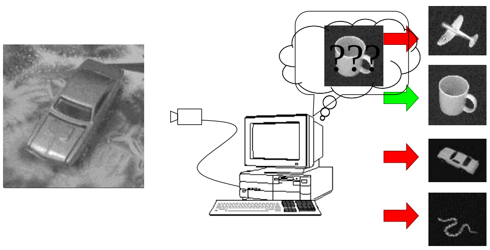

## Appearance based Recognition

- With multiple training views





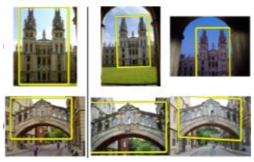

## Identification vs. Categorization






## Identification vs. Categorization

- Find *this particular* object



- Recognize ANY car

- Recognize ANY cow



## Object Categorization - Potential Applications

There is a wide range of applications, including…

Autonomous robots

Navigation, driver safety

Consumer electronics

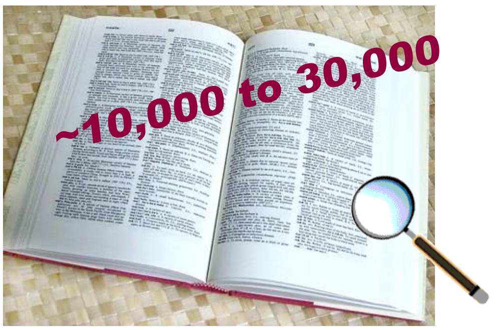

Content-based retrieval and analysis for images and videos

Medical image analysis



Slide adapted from Kristen Grauman

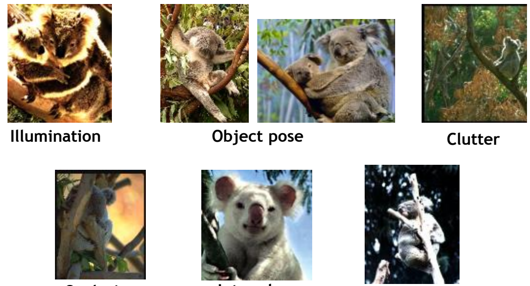
## How many object categories are there?




Source: Fei-Fei Li, Rob Fergus, Antonio Torralba.

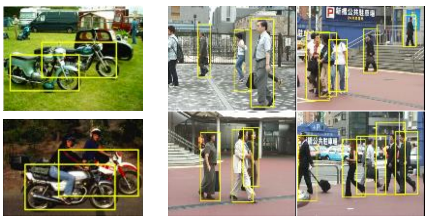
Biederman 1987


## Challenges: Robustness






## Challenges: Robustness



- Detection in crowded, real-world scenes
  - Learn object variability
    - Changes in appearance, scale, and articulation
  - Compensate for clutter, overlap, and occlusion


[Leibe, Seemann, Schiele, CVPR'05]

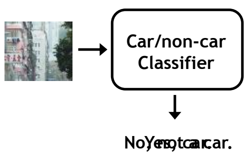
## Topics of This Lecture

- Object Categorization
  - Problem Definition
  - Challenges
- Sliding-Window based Object Detection
  - Detection via Classification
  - Global Representations
  - Classifier Construction
- Classification with SVMs
  - Support Vector Machines
  - HOG Detector
- Classification with Boosting
  - AdaBoost
  - Viola-Jones Face Detection



## Detection via Classification: Main Idea

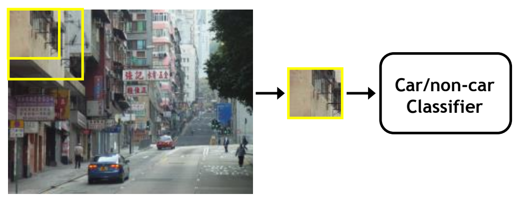
- Basic component: a binary classifier






## Detection via Classification: Main Idea

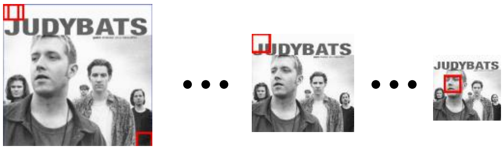
- If the object may be in a cluttered scene, slide a window around looking for it.




- Essentially, this is a brute-force approach with many local decisions.

# What *is* a Sliding Window Approach?

- Search over space and scale



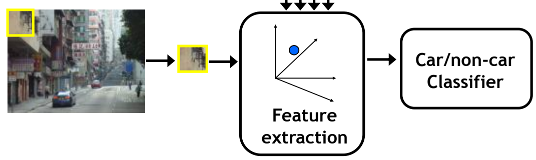
- Detection as subwindow classification problem
- *"In the absence of a more intelligent strategy, any global image classification approach can be converted into a localization approach by using a sliding-window search."*



# Detection via Classification: Main Idea

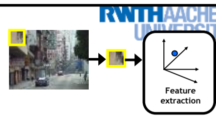
Fleshing out this pipeline a bit more, we need to:

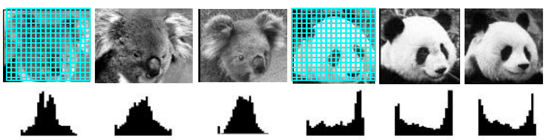
1. Obtain training data
2. Define features
3. Define classifier




# Feature extraction: Global Appearance



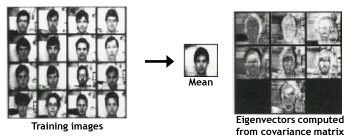
Simple holistic descriptions of image content

- Grayscale / color histogram
- Vector of pixel intensities

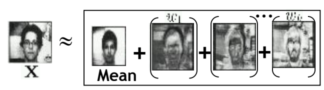


# Eigenfaces: Global Appearance Description

This can also be applied in a sliding-window framework…




Generate low-dimensional representation of appearance with a linear subspace.




Project new images to "face space".

Detection via distance TO eigenspace

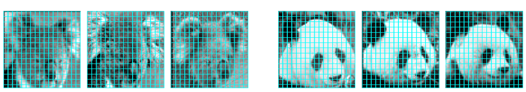
Identification via distance IN eigenspace

[Turk & Pentland, 1991]



# Feature Extraction: Global Appearance

- Pixel-based representations are sensitive to small shifts



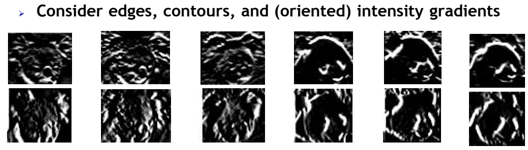
- Color or grayscale-based appearance description can be sensitive to illumination and intra-class appearance variation

Cartoon example: an albino koala



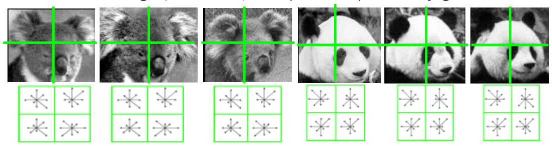
# Gradient-based Representations

- Idea
  - Consider edges, contours, and (oriented) intensity gradients



## Gradient-based Representations

- Idea
  - Consider edges, contours, and (oriented) intensity gradients

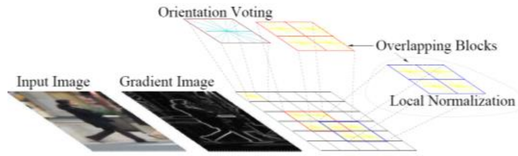


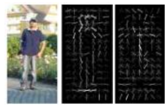
- Summarize local distribution of gradients with histogram
  - Locally orderless: offers invariance to small shifts and rotations
  - Localized histograms offer more spatial information than a single global histogram (tradeoff invariant vs. discriminative)
  - Contrast-normalization: try to correct for variable illumination

Slide credit: Kristen Grauman



## Gradient-based Representations: Histograms of Oriented Gradients (HoG)



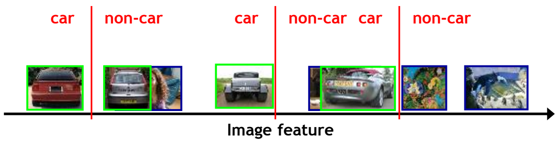
- Map each grid cell in the input window to a histogram counting the gradients per orientation.
- Code available: http://pascal.inrialpes.fr/soft/olt/

Slide credit: Kristen Grauman [Dalal & Triggs, CVPR 2005]



## Classifier Construction

- How to compute a decision for each subwindow?

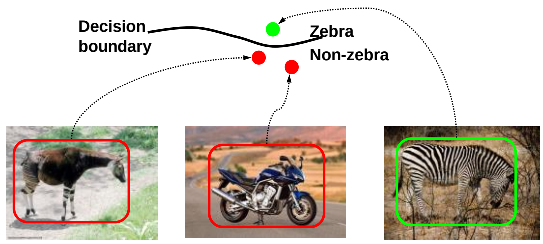



Slide credit: Kristen Grauman



## Discriminative Methods

- Learn a decision rule (classifier) assigning image features to different classes




Slide adapted from Svetlana Lazebnik

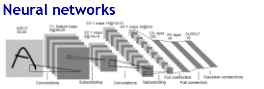


## Classifier Construction: Many Choices…

**Nearest Neighbor**

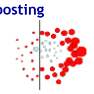
Berg, Berg, Malik 2005, Chum, Zisserman 2007, Boiman, Shechtman, Irani 2008, …

**Neural networks**



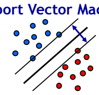
LeCun, Bottou, Bengio, Haffner 1998 Rowley, Baluja, Kanade 1998 …

**Boosting**



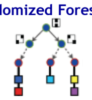
Viola, Jones 2001, Torralba et al. 2004, Opelt et al. 2006, Benenson 2012, …

**Support Vector Machines**



Vapnik, Schölkopf 1995, Papageorgiou, Poggio '01, Dalal, Triggs 2005, Vedaldi, Zisserman 2012

**Randomized Forests**

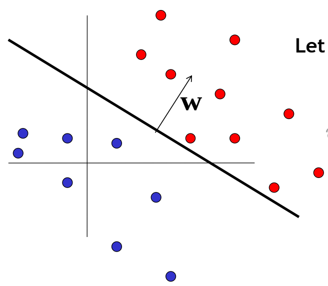


Amit, Geman 1997, Breiman 2001, Lepetit, Fua 2006, Gall, Lempitsky 2009,…

Slide adapted from Kristen Grauman



## Linear Classifiers




Let $\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

$$w_1 x_1 + w_2 x_2 + b = 0$$

$$\Updownarrow$$

$$\mathbf{w}^T \mathbf{x} + b = 0$$

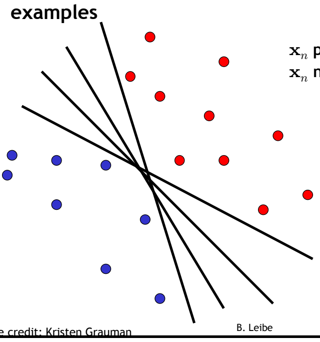
Slide adapted from: Kristen Grauman

## Linear Classifiers



- Find linear function to separate positive and negative examples

$\mathbf{x}_n$ positive: $\mathbf{w}^T\mathbf{x}_n + b \geq 0$
$\mathbf{x}_n$ negative: $\mathbf{w}^T\mathbf{x}_n + b < 0$

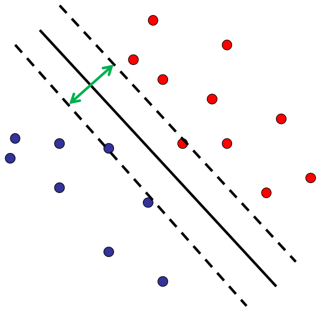


Which line is best?



## Support Vector Machines (SVMs)





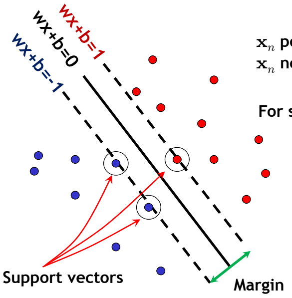
- Discriminative classifier based on *optimal separating hyperplane (i.e. line for 2D case)*
- Maximize the ***margin*** between the positive and negative training examples



## Support Vector Machines



- Want line that maximizes the margin.




$\mathbf{x}_n$ positive $(t_n = 1)$: $\mathbf{w}^T\mathbf{x}_n + b \geq 1$
$\mathbf{x}_n$ negative $(t_n = -1)$: $\mathbf{w}^T\mathbf{x}_n + b < -1$

For support, vectors, $\mathbf{w}^T\mathbf{x}_n + b = \pm 1$

*Quadratic optimization problem*

Minimize $\frac{1}{2}\mathbf{w}^T\mathbf{w}$
Subject to $t_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$

Packages available for that…

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998



## Finding the Maximum Margin Line



- Solution: $\mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n$

Learned weight
Support vector

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998



## Finding the Maximum Margin Line



- Solution: $\mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n$
- Classification function:

$$
\begin{aligned}
f(\mathbf{x}) &= \operatorname{sign}(\mathbf{w}^T\mathbf{x} + b) \\
&= \operatorname{sign}\left(\sum_{n=1}^{N} a_n t_n \boxed{\mathbf{x}_n^T\mathbf{x}} + b\right)
\end{aligned}
$$

*If f(x) < 0, classify as neg., if f(x) > 0, classify as pos.*

  - Notice that this relies on an *inner product* between the test point $\mathbf{x}$ and the support vectors $\mathbf{x}_n$
  - (Solving the optimization problem also involves computing the inner products $\mathbf{x}_n^T\mathbf{x}_m$ between all pairs of training points)

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998



## Questions



- What if the features are not 2d?
- What if the data is not linearly separable?
- What if we have more than just two categories?

## Questions

- What if the features are not 2d?
  - Generalizes to d-dimensions - replace line with “hyperplane”
- What if the data is not linearly separable?
- What if we have more than just two categories?



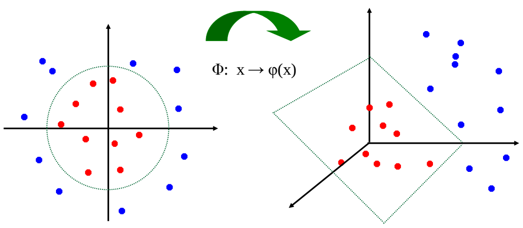
## Questions

- What if the features are not 2d?
  - Generalizes to d-dimensions - replace line with “hyperplane”
- What if the data is not linearly separable?
  - Non-linear SVMs with special kernels
- What if we have more than just two categories?



## Non-Linear SVMs: Feature Spaces

- General idea: The original input space can be mapped to some higher-dimensional feature space where the training set is separable:




More on that in the Machine Learning lecture…



## Nonlinear SVMs

- *The kernel trick*: instead of explicitly computing the lifting transformation $\varphi(\mathbf{x})$, define a kernel function K such that

$$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$$

- This gives a nonlinear decision boundary in the original feature space:

$$\sum_n a_n t_n K(\mathbf{x}_n, \mathbf{x}) + b$$

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998



## Some Often-Used Kernel Functions

- Linear:

$$K(x_i, x_j) = x_i^T x_j$$

- Polynomial of power p:

$$K(x_i, x_j) = (1 + x_i^T x_j)^p$$

- Gaussian (Radial-Basis Function):

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$



## Questions

- What if the features are not 2d?
  - Generalizes to d-dimensions - replace line with “hyperplane”
- What if the data is not linearly separable?
  - Non-linear SVMs with special kernels
- What if we have more than just two categories?

## Multi-Class SVMs



- Achieve multi-class classifier by combining a number of binary classifiers
- One vs. all
  - Training: learn an SVM for each class vs. the rest
  - Testing: apply each SVM to test example and assign to it the class of the SVM that returns the highest decision value
- One vs. one
  - Training: learn an SVM for each pair of classes
  - Testing: each learned SVM “votes” for a class to assign to the test example



## SVMs for Recognition



1. Define your representation for each example.
2. Select a kernel function.
3. Compute pairwise kernel values between labeled examples
4. Pass this “kernel matrix” to SVM optimization software to identify support vectors & weights.
5. To classify a new example: compute kernel values between new input and support vectors, apply weights, check sign of output.




## Topics of This Lecture



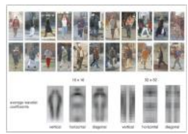
- Object Categorization
  - Problem Definition
  - Challenges
- Sliding-Window based Object Detection
  - Detection via Classification
  - Global Representations
  - Classifier Construction
- Classification with SVMs
  - Support Vector Machines
  - HOG Detector
- Classification with Boosting
  - AdaBoost
  - Viola-Jones Face Detection



## Pedestrian Detection

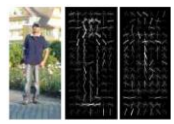


- Detecting upright, walking humans using sliding window’s appearance/texture; e.g.,



SVM with Haar wavelets [Papageorgiou & Poggio, IJCV 2000]

Space-time rectangle features [Viola, Jones & Snow, ICCV 2003]



SVM with HoGs [Dalal & Triggs, CVPR 2005]



## HOG Descriptor Processing Chain



Image Window



## HOG Descriptor Processing Chain



- Optional: Gamma compression
  - Goal: Reduce effect of overly strong gradients
  - Replace each pixel color/intensity by its square-root

$$x \mapsto \sqrt{x}$$

$\Rightarrow$ Small performance improvement

Gamma compression

↑

Image Window

## HOG Descriptor Processing Chain

- **Gradient computation**
  - Compute gradients on all color channels and take strongest one
  - Simple finite difference filters work best (no Gaussian smoothing)

$$\begin{bmatrix}-1 & 0 & 1\end{bmatrix} \qquad \begin{bmatrix}-1\\0\\1\end{bmatrix}$$

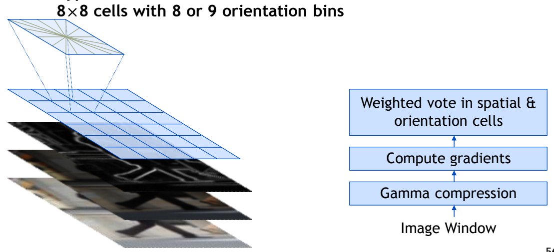

Computer Vision WS 16/17

Slide adapted from Navneet Dalal

55

## HOG Descriptor Processing Chain

- **Spatial/Orientation binning**
  - Compute localized histograms of oriented gradients
  - Typical subdivision: 8×8 cells with 8 or 9 orientation bins



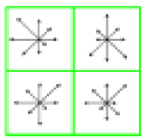

Computer Vision WS 16/17

Slide adapted from Navneet Dalal

56

## HOG Cell Computation Details

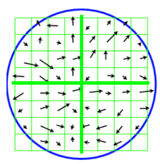
- **Gradient orientation voting**
  - Each pixel contributes to localized gradient orientation histogram(s)
  - Vote is weighted by the pixel's gradient magnitude



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

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

- **Block-level Gaussian weighting**
  - An additional Gaussian weight is applied to each 2×2 block of cells
  - Each cell is part of 4 such blocks, resulting in 4 versions of the histogram.

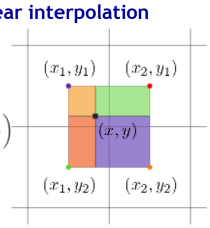


Computer Vision WS 16/17

57

## HOG Cell Computation Details (2)

- **Important for robustness: Tri-linear interpolation**
  - Each pixel contributes to (up to) 4 neighboring cell histograms
  - Weights are obtained by bilinear interpolation in image space:



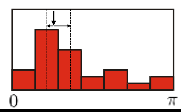

$$h(x_1,y_1) \leftarrow w \cdot \left(1 - \frac{x - x_1}{x_2 - x_1}\right)\left(1 - \frac{y - y_1}{y_2 - y_1}\right)$$

$$h(x_1,y_2) \leftarrow w \cdot \left(1 - \frac{x - x_1}{x_2 - x_1}\right)\left(\frac{y - y_1}{y_2 - y_1}\right)$$

$$h(x_2,y_1) \leftarrow w \cdot \left(\frac{x - x_1}{x_2 - x_1}\right)\left(1 - \frac{y - y_1}{y_2 - y_1}\right)$$

$$h(x_2,y_2) \leftarrow w \cdot \left(\frac{x - x_1}{x_2 - x_1}\right)\left(\frac{y - y_1}{y_2 - y_1}\right)$$

  - Contribution is further split over (up to) 2 neighboring orientation bins via linear interpolation over angles.

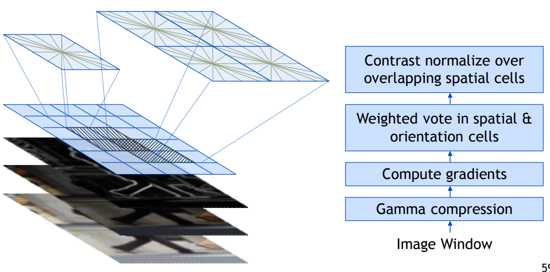



Computer Vision WS 16/17

58

## HOG Descriptor Processing Chain

- **2-Stage contrast normalization**
  - L2 normalization, clipping, L2 normalization




Computer Vision WS 16/17

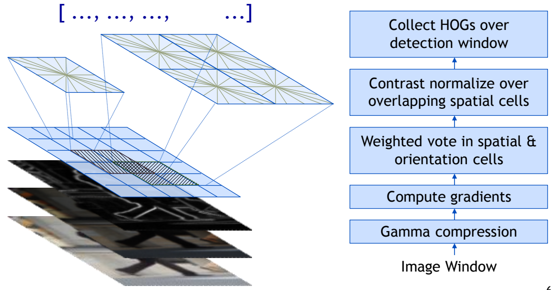
Slide adapted from Navneet Dalal

59

## HOG Descriptor Processing Chain

- **Feature vector construction**
  - Collect HOG blocks into vector

[ …, …, …, …]




Computer Vision WS 16/17

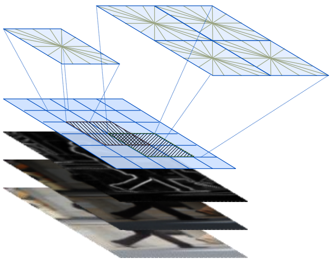
Slide adapted from Navneet Dalal

60

## HOG Descriptor Processing Chain

- SVM Classification
  - Typically using a linear SVM

[ ..., ..., ..., ...]



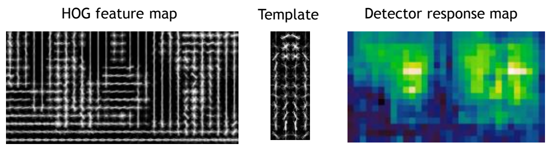
Object/Non-object
↑
Linear SVM
↑
Collect HOGs over detection window
↑
Contrast normalize over overlapping spatial cells
↑
Weighted vote in spatial & orientation cells
↑
Compute gradients
↑
Gamma compression
↑
Image Window



Slide adapted from Navneet Dalal

## Pedestrian Detection with HOG

- Train a pedestrian template using a linear SVM
- At test time, convolve feature map with template




N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005



Slide credit: Svetlana Lazebnik

## Non-Maximum Suppression

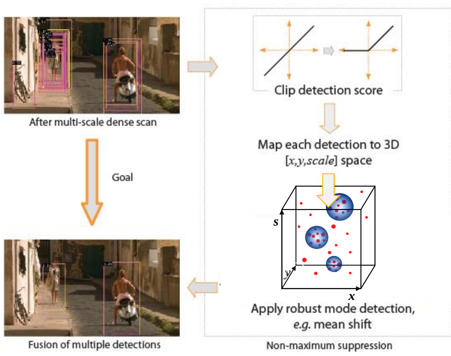





Image source: Navneet Dalal, PhD Thesis

## Pedestrian detection with HoGs & SVMs

- Navneet Dalal, Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005



Slide credit: Kristen Grauman

## References and Further Reading

- Read the HOG paper
  - N. Dalal, B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR, 2005.
- HOG Detector
  - Code available: http://pascal.inrialpes.fr/soft/olt/