Computer Vision - Lecture 9

Sliding-Window based Object Detection

26.11.2015

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
  - Global Representations
  - Sliding-Window Object Detection
  - Image Classification

• Local Features & Matching

• 3D Reconstruction

• Motion and Tracking
Recap: 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
Recap: Recognition Using Histograms

• Histogram comparison

Test image

Known objects
Recap: Comparison Measures

- Vector space interpretation
  - Euclidean distance
  - Mahalanobis distance

- Statistical motivation
  - Chi-square
  - Bhattacharyya

- Information-theoretic motivation
  - Kullback-Leibler divergence, Jeffreys divergence

- Histogram motivation
  - Histogram intersection

- Ground distance
  - Earth Movers Distance (EMD)
Recap: Recognition Using Histograms

- Simple algorithm
  1. Build a set of histograms $H=\{h_i\}$ for each known object
     - More exactly, for each view of each object
  2. Build a histogram $h_t$ for the test image.
  3. Compare $h_t$ to each $h_i \in H$
     - Using a suitable comparison measure
  4. Select the object with the best matching score
     - Or reject the test image if no object is similar enough.

“Nearest-Neighbor” strategy
Recap: Multidimensional Representations

- Combination of several descriptors
  - Each descriptor is applied to the whole image.
  - Corresponding pixel values are combined into one feature vector.
  - Feature vectors are collected in multidimensional histogram.
Application: Brand Identification in Video

[Hall, Pellison, Riff, Crowley, 2004]
Application: Brand Identification in Video

B. Leibe

[Hall, Pellison, Riff, Crowley, 2004]
Application: Brand Identification in Video

false detection

<table>
<thead>
<tr>
<th>Brand</th>
<th>Detection Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agip</td>
<td>2%</td>
</tr>
<tr>
<td>Aral. Alles super.</td>
<td>3%</td>
</tr>
<tr>
<td>Fosters</td>
<td>11%</td>
</tr>
<tr>
<td>Helix</td>
<td>0%</td>
</tr>
<tr>
<td>Marlboro</td>
<td>33%</td>
</tr>
</tbody>
</table>
You’re Now Ready for First Applications…

- Binary Segmentation
- Moment descriptors
- Skin color detection
- Circle detection
- Line detection
- Histogram based recognition
- Binary Segmentation

Image Source: http://www.flickr.com/photos/angelsk/2806412807/
Topics of This Lecture

- **Object Categorization**
  - Problem Definition
  - Challenges

- **Sliding-Window based Object Detection**
  - Detection via Classification
  - Global Representations
  - Classifier Construction

- **Classification with Boosting**
  - AdaBoost
  - Viola-Jones Face Detection

- **Classification with SVMs**
  - Support Vector Machines
  - HOG Detector
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
- Medical image analysis

Content-based retrieval and analysis for images and videos

Slide adapted from Kristen Grauman
How many object categories are there?

~10,000 to 30,000

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

Biederman 1987
Challenges: Robustness

- Illumination
- Object pose
- Clutter
- Occlusions
- Intra-class appearance
- Viewpoint

Slide credit: Kristen Grauman
Challenges: Robustness

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

B. Leibe

[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 Boosting**
  - AdaBoost
  - Viola-Jones Face Detection

- **Classification with SVMs**
  - Support Vector Machines
  - HOG Detector
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.

Slide credit: Kristen Grauman
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.”

B. Leibe
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

Slide credit: Kristen Grauman
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...

Training images

Eigenvectors computed from covariance matrix

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

\[ \mathbf{X} \approx \text{Mean} + \sum_{k} w_k \]

Slide credit: Kristen Grauman

B. Leibe

[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

Slide credit: Kristen Grauman
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.


[Dalal & Triggs, CVPR 2005]
Classifier Construction

- How to compute a decision for each subwindow?
**Discriminative Methods**

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

---

Slide adapted from Svetlana Lazebnik

B. Leibe
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


Randomized Forests

Linear Classifiers

Let \( w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \) \( x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \)

\[
\begin{align*}
w_1 x_1 + w_2 x_2 + b &= 0 \\
w^T x + b &= 0
\end{align*}
\]
Linear Classifiers

- Find linear function to separate positive and negative examples

\[ x_n \text{ positive: } w^T x_n + b \geq 0 \]
\[ x_n \text{ negative: } w^T x_n + b < 0 \]
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

Slide credit: Kristen Grauman
Support Vector Machines

- Want line that maximizes the margin.

\[ \begin{align*}
\text{x}_n \text{ positive (} t_n = 1\text{):} & \quad w^T x_n + b \geq 1 \\
\text{x}_n \text{ negative (} t_n = -1\text{):} & \quad w^T x_n + b < -1
\end{align*} \]

For support vectors, \( w^T x_n + b = \pm 1 \)

**Quadratic optimization problem**

Minimize \( \frac{1}{2} w^T w \)

Subject to \( t_n (w^T x_n + b) \geq 1 \)

Finding the Maximum Margin Line

• Solution: \[ w = \sum_{n=1}^{N} a_n t_n x_n \]

Finding the Maximum Margin Line

• Solution: \( \mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n \)

• Classification function:

\[
f(x) = \text{sign}(\mathbf{w}^T \mathbf{x} + b)
\]

\[
= \text{sign}\left(\sum_{n=1}^{N} a_n t_n \mathbf{x}_n^T \mathbf{x} + b\right)
\]

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)

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:

\[ \Phi: x \rightarrow \varphi(x) \]

More on that in the Machine Learning lecture...

Slide from Andrew Moore’s tutorial: [http://www.autonlab.org/tutorials/svm.html](http://www.autonlab.org/tutorials/svm.html)
Nonlinear SVMs

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

$$K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$$

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

$$\sum_n a_n t_n K(x_n, x) + b$$

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):**
  \[ K(x_i,x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \]

Slide from Andrew Moore’s tutorial: [http://www.autonlab.org/tutorials/svm.html](http://www.autonlab.org/tutorials/svm.html)
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

Slide credit: Kristen Grauman
SVMs for Recognition

1. Define your representation for each example.

2. Select a kernel function.

3. Compute pairwise kernel values between labeled examples.

4. Given 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.

Slide credit: Kristen Grauman
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

Slide adapted from Navneet Dalal
HOG Descriptor Processing Chain

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

\[ x \mapsto \sqrt{x} \]

⇒ Small performance improvement
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)

\[
\begin{bmatrix}
-1 & 0 & 1
\end{bmatrix} \quad \begin{bmatrix}
-1 \\
0 \\
1
\end{bmatrix}
\]

Slide adapted from Navneet Dalal
HOG Descriptor Processing Chain

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

Image Window

Weighted vote in spatial & orientation cells

Compute gradients

Gamma compression

Image Window

Slide adapted from Navneet Dalal
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

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

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

- **Block-level Gaussian weighting**
  - An additional Gaussian weight is applied to each $2 \times 2$ block of cells
  - Each cell is part of 4 such blocks, resulting in 4 versions of the histogram.
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:
    
    \[
    \begin{align*}
    h(x_1, y_1) &\leftarrow w \cdot \left(1 - \frac{x - x_1}{x_2 - x_1}\right) \left(1 - \frac{y - y_1}{y_2 - y_1}\right) \\
    h(x_1, y_2) &\leftarrow w \cdot \left(1 - \frac{x - x_1}{x_2 - x_1}\right) \left(\frac{y - y_1}{y_2 - y_1}\right) \\
    h(x_2, y_1) &\leftarrow w \cdot \left(\frac{x - x_1}{x_2 - x_1}\right) \left(1 - \frac{y - y_1}{y_2 - y_1}\right) \\
    h(x_2, y_2) &\leftarrow w \cdot \left(\frac{x - x_1}{x_2 - x_1}\right) \left(\frac{y - y_1}{y_2 - y_1}\right)
    \end{align*}
    \]

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

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

Slide adapted from Navneet Dalal
HOG Descriptor Processing Chain

- Feature vector construction
  - Collect HOG blocks into vector
  
    \[
    [\ldots,\ldots,\ldots,\ldots]\n    \]

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
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


Slide credit: Svetlana Lazebnik
Non-Maximum Suppression

After multi-scale dense scan

Clip detection score

Map each detection to 3D [x,y, scale] space

Apply robust mode detection, e.g. mean shift

Non-maximum suppression

Fusion of multiple detections

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

• HOG Detector
  - Code available: http://pascal.inrialpes.fr/soft/olt/