Computer Vision - Lecture 10

Sliding-Window based Object Detection

29.11.2011

Bastian Leibe
RWTH Aachen
http://www.mmp.rwth-aachen.de
leibe@umic.rwth-aachen.de

Recap: Subspace Methods

Subspace methods

Reconstructive
PCA, ICA, NMF

Discriminative
FLD, SVM, CCA

representation
classification
regression

Recap: Obj. Detection by Distance TO Eigenspace

- For each test image, compute the reprojection error
  - An n-pixel image in \( R^n \) can be projected to the low-dimensional feature space \( \psi(x) \) by \( y = Ux \).
  - From \( y \in R^m \), the reconstruction of the point is \( U^Ty \).
  - The error of the reconstruction is \( \| y - U^TUs \| \).
  - Accept a detection if this error is low.
    - Assumption: subspace is optimized to the target object (class).
    - Other classes are not represented well \( \Rightarrow \) large error.

Recap: Obj Identification by Distance IN Eigenspace

- Objects are represented as coordinates in an \( n \)-dim. eigenspace.
- Example:
  - 3D space with points representing individual objects or a manifold representing parametric eigenspace (e.g., orientation, pose, illumination).
- Estimate parameters by finding the NN in the eigenspace.

Recap: Eigenfaces

Course Outline

- Image Processing Basics
- Segmentation & Grouping
- Recognition
  - Global Representations
  - Subspace Representations
- Object Categorization I
  - Sliding Window based Object Detection
- Local Features & Matching
- Object Categorization II
  - Part based Approaches
- 3D Reconstruction
- Motion and Tracking
Recap: Restrictions of PCA

- PCA minimizes projection error
- PCA is “unsupervised” no information on classes is used
- Discriminating information might be lost

Recap: Linear Discriminant Analysis (FLD)

- Maximize distance between classes
- Minimize distance within a class
- Criterion: \( J(w) = \frac{w^T S_B w}{w^T S_W w} \)
- In the two-class case, the optimal solution for \( w \) can be obtained as:
  \[ w \propto S_W^{-1}(m_2 - m_1) \]
- Classification function:
  \[ y(x) = w^T x + w_0 \]

Recap: Multiple Discriminant Analysis

- Generalization to \( K \) classes
  \[ J(W) = \frac{|W^T S_B W|}{|W^T S_W W|} \]
  \( S_B w_i = \lambda_i S_W w_i \)
  \( \text{Solution given by generalized eigenvalue problem} \)
  - Defining \( v = S_W^{-\frac{1}{2}} w \), we get
    \[ S_B S_W^{-1} S_B^{-1} v = \lambda v \]
    which is a regular eigenvalue problem.
    \( \Rightarrow \text{Solve to get eigenvectors of } v \), then from that of \( w \).  
    - For the \( K \)-class case we obtain (at most) \( K-1 \) projections.
      - (i.e. eigenvectors corresponding to non-zero eigenvalues.)

Recap: Fisherfaces

- Example Fisherface for recognition “Glasses/NoGlasses”

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

**Object Categorization**

- Task Description
  - “Given a small number of training images of a category, recognize a-priori unknown instances of that category and assign the correct category label.”
- Which categories are feasible visually?
  - Extensively studied in Cognitive Psychology, e.g. [Brown’58]

**Visual Object Categories**

- Basic-Level Categories in human categorization
  - The highest level at which category members have similar perceived shape
  - The highest level at which a single mental image reflects the entire category
  - The level at which human subjects are usually fastest at identifying category members
  - The first level named and understood by children
  - The highest level at which a person uses similar motor actions for interaction with category members

**How many object categories are there?**

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

Source: Fan-Fat Li, Rob Fergus, Antonio Torralba.
Other Types of Categories

- Functional Categories
  - e.g. chairs = "something you can sit on"

Challenges: Robustness

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

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 object may be in a cluttered scene, slide a window around looking for it.
  ![Car/non-car Classifier](image)
• Essentially, this is a brute-force approach with many local decisions.

What is a Sliding Window Approach?
• Search over space and scale
  ![Car/Non-car](image)
• 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".

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](image)
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
  - Still more spatial information than single global histogram
  - Contrast-normalization: try to correct for variable illumination

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

- Code available: [http://pascal.inrialpes.fr/software/]

Classifier Construction

- How to compute a decision for each subwindow?

Discriminative Methods

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

Classifier Construction: Many Choices...

Nearest neighbor

- Shahnaarovitch, Viola, Darrell 2003
- Berg, Berg, Malik 2005...

Support Vector Machines

- Guyon, Vapnik, Heisele, Serre, Poggio, 2001...
- viola, Jones 2001
- Terraila et al. 2004
- Opelt et al. 2006...

Boosting

- LeCun, Bottou, Bengio, Haffner 1998
- Rowley, Baluja, Kanade 1998...

Conditional Random Fields

- McCallum, Freitag, Pereira 2000
- Kumar, Hebert 2003...

Support Vector Machines

- LeCun, Bottou, Bengio, Haffner 1998
- Rowley, Baluja, Kanade 1998...

Boosting

- McCallum, Freitag, Pereira 2000
- Kumar, Hebert 2003...

Conditional Random Fields

- McCallum, Freitag, Pereira 2000
- Kumar, Hebert 2003...
AdaBoost: Intuition

Consider a 2D feature space with positive and negative examples.
Each weak classifier splits the training examples with at least 50% accuracy.
Examples misclassified by a previous weak learner are given more emphasis at future rounds.

AdaBoost: Basic Steps

1. Start with uniform weighting on training examples
2. For M iterations
   a) Select best weak classifier for this weighted data set:
      - Evaluate weighted error for each feature, pick best one.
   b) Reweight the training examples:
      - Incorrectly classified ⇒ more weight
      - Correctly classified ⇒ less weight
3. Final classifier is combination of the weak ones, weighted according to the error they had.
4. Simple, eh? Let’s flesh this out in more detail...

AdaBoost: Detailed Training Algorithm

1. Initialization: Set $w_0(x) = \frac{1}{N}$ for $n = 1, \ldots, N$.
2. For $m = 1, \ldots, M$ iterations
   a) Train a new weak classifier $h_m(x)$ using the current weighting coefficients $W^m$ by minimizing the weighted error function
      $$J_m = \sum_{x \in X} w_m(x) I(h_m(x) \neq t_x)$$
   b) Estimate the weighted error of this classifier on $X$:
      $$\epsilon_m = \frac{\sum_{x \in X} w_m(x) I(h_m(x) \neq t_x)}{\sum_{x \in X} w_m(x)}$$
   c) Calculate a weighting coefficient for $h_m(x)$:
      $$\alpha_m = \ln \frac{1 - \epsilon_m}{\epsilon_m}$$
   d) Update the weighting coefficients:
      $$w_{m+1}(x) = w_m(x) \exp \{\alpha_m I(h_m(x) \neq t_x)\}$$
AdaBoost: Recognition

- Evaluate all selected weak classifiers on test data.
  \[ h_1(x), \ldots, h_m(x) \]
- Final classifier is weighted combination of selected weak classifiers:
  \[ H(x) = \text{sign} \left( \sum_{m=1}^{M} \alpha_m h_m(x) \right) \]
- Very simple procedure!
  - Less than 10 lines in Matlab!
  - But works extremely well in practice...

Example: Face Detection

- Frontal faces are a good example of a class where global appearance models + a sliding window detection approach fit well:
  - Regular 2D structure
  - Center of face almost shaped like a “patch”/window
- Now we’ll take AdaBoost and see how the Viola-Jones face detector works

Feature extraction

“Rectangular” filters

Feature output is difference between adjacent regions

Efficiently computable with integral image: any sum can be computed in constant time

Avoid scaling images ➔ scale features directly for same cost

Large Library of Filters

Considering all possible filter parameters: position, scale, and type:
180,000+ possible features associated with each 24 x 24 window

Use AdaBoost both to select the informative features and to form the classifier

Weak classifier: filter output > 0?

AdaBoost for Feature+Classifier Selection

- Want to select the single rectangle feature and threshold that best separates positive (faces) and negative (non-faces) training examples, in terms of weighted error.

Resulting weak classifier:

\[ h(x) = \begin{cases} +1 & \text{if } f(x) > 0; \\ -1 & \text{otherwise} \end{cases} \]

For next round, reweight the examples according to errors, choose another filter/threshold combo.
AdaBoost for Efficient Feature Selection

- Image features = weak classifiers
- For each round of boosting:
  - Evaluate each rectangle filter on each example
  - Sort examples by filter values
  - Select best threshold for each filter (min error)
  - Sorted list can be quickly scanned for the optimal threshold
  - Select best filter/threshold combination
  - Weight on this features is a simple function of error rate
  - Reweight examples


Cascading Classifiers for Detection

- Even if the filters are fast to compute, each new image has a lot of possible windows to search.
- For efficiency, apply less accurate but faster classifiers first to immediately discard windows that clearly appear to be negative; e.g.,
  - Filter for promising regions with an initial inexpensive classifier
  - Build a chain of classifiers, choosing cheap ones with low false negative rates early in the chain

Viola-Jones Face Detector: Summary

- Train with 5K positives, 350M negatives
- Real-time detector using 38 layer cascade
- 6061 features in final layer
  - Implementation available in OpenCV:
    - http://sourceforge.net/projects/opencvlibrary/

Viola-Jones Face Detector: Results

- First two features selected
- Performance
  - 384 by 288 pixel images detected at 15 fps on a conventional 700 MHz Intel Pentium III in 2001.
  - Training time = weeks

Viola-Jones Face Detector: Results
Viola-Jones Face Detector: Results

Perceptual and Sensory Augmented Computing
Computer Vision WS 11/12

Viola-Jones Face Detector: Results

Perceptual and Sensory Augmented Computing
Computer Vision WS 11/12

You Can Try It At Home...

• The Viola & Jones detector was a huge success
  ▶ First real-time face detector available
  ▶ Many derivative works and improvements

• C++ implementation available in OpenCV [Lienhart, 2002]
  ▶ http://sourceforge.net/projects/opencvlibrary/
• Matlab wrappers for OpenCV code available, e.g. here
  ▶ http://www.mathworks.com/matlabcentral/fileexchange/19912

P. Viola, M. Jones, Robust Real-Time Face Detection, IJCV, Vol. 57(2), 2004

Example Application

Frontal faces detected and then tracked, character names inferred with alignment of script and subtitles.

Everingham, M., Sivic, J. and Zisserman, A. "Hello! My name is... Buffy" - Automatic naming of characters in TV video, BMVC 2006.
http://www.robots.ox.ac.uk/~vgg/research/face/index.html

Classifier Construction: Many Choices...

Linear Classifiers

Let \[ w = \begin{bmatrix} a \\ c \end{bmatrix} \quad x = \begin{bmatrix} x \\ y \end{bmatrix} \]
\[ ax + cy + b = 0 \]
\[ \mathbf{w} \cdot \mathbf{x} + b = 0 \]
**Linear Classifiers**

- Find linear function to separate positive and negative examples

}\[ x, \text{ positive: } x \cdot w + b \geq 0 \]
\[ x, \text{ negative: } x \cdot w + b < 0 \]

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.

\[ x, \text{ positive (} y = 1\): } x \cdot w + b \geq 1 \]
\[ x, \text{ negative (} y = -1\): } x \cdot w + b \leq -1 \]

For support vectors, \( x \cdot w + b = \pm 1 \)

**Finding the Maximum Margin Line**

- Solution:

\[ w = \sum \alpha_i y_i x_i \]

- Classification function:

\[ f(x) = \text{sign}(w \cdot x + b) \]

- Notice that this relies on an inner product between the test point \( x \) and the support vectors \( x_i \)
- (Solving the optimization problem also involves computing the inner products \( x_i \cdot x_j \) 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?

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...

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

Nonlinear SVMs

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

\[ K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \]

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

\[ \sum_i \alpha_i y_i K(x_i, x) + b \]

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

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]

Pedestrian detection with HoGs & SVMs


Summary: Sliding-Windows

- **Pros**
  - Simple detection protocol to implement
  - Good feature choices critical
  - Past successes for certain classes
  - Good detectors available (Viola & Jones, HOG, etc.)
- **Cons/Limitations**
  - High computational complexity
    - For example: 250,000 locations x 30 orientations x 4 scales = 30,000,000 evaluations!
    - This puts tight constraints on the classifiers we can use.
    - If training binary detectors independently, this means cost increases linearly with number of classes.
  - With so many windows, false positive rate better be low

Limitations (continued)

- Not all objects are “box” shaped
Limitations (continued)

- Non-rigid, deformable objects not captured well with representations assuming a fixed 2D structure; or must assume fixed viewpoint
- Objects with less-regular textures not captured well with holistic appearance-based descriptions

Limitations (continued)

- If considering windows in isolation, context is lost

References and Further Reading

- Read the Viola-Jones paper
  - (first version appeared at CVPR 2001)

- Viola-Jones Face Detector
  - C++ implementation available in OpenCV [Lienhart, 2002]
    - http://sourceforge.net/projects/opencvlibrary/
  - Matlab wrappers for OpenCV code available, e.g. here

- HOG Detector
  - Code available: http://pascal.inrialpes.fr/software/alt/