CS231M · Mobile Computer Vision

Announcements

- Next Wed team presentations start
- Please select the paper you want to present
- P2 submission deadline has been postponed to Friday 16th
Recognition

- Classification
- Detection
- Single instance detection and localization
Feature Detection

- Estimation
- Matching
- Indexing
- Detection

e.g. DoG

Feature Description

e.g. SIFT

From low level to high level vision
Classification or indexing

Is this an image of a bridge?
Image search engines
Detection

Does this image contain a bridge? [where?]
Face detection
Human body detection and gesture recognition
Single instance detection
Does this image contain the golden gate bridge? [where?]
Or which landmark does this image contain?
Google Goggles

Visual search and landmarks recognition
Visual search and landmarks recognition
Face identification
Fingerprint identification
Challenges: illumination

image credit: J. Koenderink
Challenges: scale
Challenges: deformation
Challenges:
occlusion

Magritte, 1957

slide credit: Fei-Fei, Fergus & Torralba
Challenges: background clutter

Kilmeny Niland. 1995
Challenges: viewpoint variation

Michelangelo 1475-1564

slide credit: Fei-Fei, Fergus & Torralba
~10,000 to 30,000
Challenges: intra-class variation
Recognition paradigm

- Representation
- Learning
- recognition
Representation

- Building blocks: Sampling strategies

Interest operators

Dense, uniformly

Multiple interest operators

Randomly

Image credits: F.-F. Li, E. Nowak, J. Sivic
Representation

– Appearance only or location and appearance
Learning: Generative models

• Naïve Bayes classifier
  – Csurka Bray, Dance & Fan, 2004

• Hierarchical Bayesian topic models (e.g. pLSA and LDA)
  – Natural scene categorization: Fei-Fei et al. 2005

• 2D Part based models
  - Constellation models: Weber et al 2000; Fergus et al 2000
  - Star models: ISM (Leibe et al 05)

• 3D part based models:
Learning: Discriminative models

Nearest neighbor
Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

Support Vector Machines
Guyon, Vapnik, Heisele, Serre, Poggio...

Neural networks
LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998

Decision trees & Random forests
Dietterich 00; Amit & Geman 97
Criminisi et al. 11

Latent SVM
Felzenszwalb 00
Ramanan 03...

Boosting
Viola, Jones 2001,
Torralba et al. 2004,
Opelt et al. 2006,...
Recognition

– Recognition task: classification, detection, etc.
Recognition

– Recognition task

– Search strategy: Sliding Windows
  • Simple
  • Computational complexity \((x,y, S, \theta, N\) of classes)
    - BSW by Lampert et al 08
    - Also, Alexe, et al 10

Viola, Jones 2001,
Recognition

- Recognition task

- Search strategy: Sliding Windows
  - Simple
  - Computational complexity ($x, y, S, \theta, N$ of classes)
    - BSW by Lampert et al 08
    - Also, Alexe, et al 10

- Localization
  - Objects are not boxes

Viola, Jones 2001,
Recognition

- Recognition task

- Search strategy: Sliding Windows
  - Simple
  - Computational complexity \((x, y, S, \theta, N \text{ of classes})\)
    - BSW by Lampert et al 08
    - Also, Alexe, et al 10
  - Localization
    - Objects are not boxes
    - Prone to false positive

Non max suppression:
- Canny ’86
- Desai et al, 2009

Viola, Jones 2001,
Classification or indexing

Is this an image of a bridge?
definition of “BoW”

– Independent features
– histogram representation
1. Feature detection and description

- Detect patches
  - [Mikojaczyk and Schmid '02]
  - [Mata, Chum, Urban & Pajdla, '02]
  - [Sivic & Zisserman, '03]

- Normalize patch
- Compute SIFT descriptor
  - [Lowe'99]

Slide credit: Josef Sivic
2. Codewords dictionary formation
2. Codewords dictionary formation
2. Codewords dictionary formation

Cluster center = code word

Clustering/ vector quantization

E.g., Kmeans, see CS131A
2. Codewords dictionary formation

- Image patch examples of codewords
3. Bag of word representation

- Nearest neighbors assignment
- K-D tree search strategy
3. Bag of word representation

Codewords dictionary

frequency

codewords
Representation

1. feature detection & representation

2. codewords dictionary

3. category models
Category models

Class 1

Class N
Recognition

codewords dictionary

category models (and/or) classifiers

category decision
Discriminative models

**Nearest neighbor**

10^6 examples

Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

**Support Vector Machines**

Guyon, Vapnik, Heisele, Serre, Poggio...

**Latent SVM**

**Structural SVM**

Felzenszwalb 00
Ramanan 03...

**Neural networks**

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

**Boosting**

Viola, Jones 2001,
Torralba et al. 2004,
Opelt et al. 2006,...
Major drawback of BOW models

Don’t capture spatial information!
Spatial Pyramid Matching

Class street
Spatial Pyramid Matching

- K. Grauman and T. Darrell 2005
- S. Lazebnik et al, 2006
- D. Nister et al. 2006,
Caltech 101

Fei-Fei et al. (2004)

Pyramid matching

Caltech 101

BOW \sim 15\%

mean recognition rate per class

number of training examples per class

- Zhang, Berg, Maire & Malik (CVPR06)
- Lazebnik, Schmid & Ponce (CVPR06)
- Berg (thesis 06)
- Mutch & Lowe (CVPR06)
- Grauman & Darrell (tech report 06)
- Berg, Berg & Malik (CVPR05)
- Wang, Zhang & Fel-Fel (CVPR06)
- Holub, Welling & Perona (ICCV05)
- Serre, Wolf & Poggio (CVPR05)
- Fel-Fel, Fergus & Perona (GMBV04)
- SSD baseline
Major drawback of BOW models

- Don’t capture spatial information!

- As the number of images/classes to model increases, the dictionary size also increases
  - Computational cost of increasing the size of the vocabulary becomes very high!
Vocabulary tree

*Scalable Recognition with a Vocabulary Tree*. David Nistér and Henrik Stewénius. 2006

- Feature vectors are hierarchically clustered into a k-way tree – also called vocabulary tree
- Computational cost in the hierarchical approach is logarithmic in the number of leaf nodes.
- Vocabularies of millions ($10^6$) of codewords can be supported
  - Individual words can be made more discriminative
  - Only 10 x 6 comparisons for quantizing each descriptor
First, an initial k-means process is run on the training data, defining k cluster centers.

The training data is then partitioned into k groups, where each group consists of the descriptor vectors closest to a particular cluster center.

The same process is then recursively applied to each group of descriptor vectors, recursively defining quantization cells by splitting each quantization cell into k new parts.
Vocabulary tree

With 40,000 images in the database, the retrieval is still real-time… (in 2006 !)
Detection

Does this image contain a bridge? [where?]
Model-based detection

1. Slide a window in image
   - E.g., choose position, scale orientation

2. Compare it with a model/template
   - Compute similarity to an example object or to a summary representation

3. Compute a score for each comparison and compute non-max suppression to remove weak scores
HoG = Histogram of Oriented Gradients

- Like SIFT, but...
  - Sampled on a dense, regular grid around the object
  - Gradients are contrast normalized in overlapping blocks

In OPEN CV: `struct CV_EXPORTS HOGDescriptor`

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05
DPM = Deformable part model

• Like HOG template, but...
  – Use a star-structured part-based model made of:
    • Root filter (similar to Dalal-Triggs)
    • Set of parts and an associated deformation model

Felzenszwalb, et al., Discriminatively Trained Deformable Part Models,
http://people.cs.uchicago.edu/~pff/latent/
Object Detection

**Deformable Part Models (DPM)**
- **DPM**: Felzenszwalb, Girshick, McAllester, Ramanan 2010
- **Sparselet**: Song et al. 2012
- **Multi-Component model**: Gu et al. 2012

**Convolutional Neural Network (CNN)**
- **CNN**: LeCun, Bottou, Bengio, Haffner 1998
- **Deep CNN**: Krizhevsky, Sutskever, Hinton 2012

**Boosting**
- **Vila-Jones Detection**: 2001
- **Regionlet**: Wang et al 2013

**3D Object Detection**
- **ALM**: Yu & Savarese, 2012
- **3D²PM**: Pepik et al 2012
- **RGBD-CPMC**: Lin et al 2013
Beyond sliding windows

Selective Search:

Selectiv e Search: Sande et al 2011
segDPM: Fidler, Mottaghi, Yuille, Urtasun 2013
Single instance detection

- Does this image contain the golden gate bridge? [where?]
- Or which landmark does this image contain?
Recognizing single instances

- Representation
  - Detectors and descriptors

- Model learning & Recognition
  - Hypothesis generation
  - Model verification
Representation

Feature descriptor
SIFT, ORB, etc…
Recognition

Goal: given a query image $I$, match objects in the image against a collection of learnt object models
**Recognition**

**Goal:** given a query image $I$, match objects in the image against a collection of learnt object models

- Match features between query image $I$ and object model
- Generate hypothesis with a few matches
- Verify hypothesis with all the remaining matches
- Select hypothesis with lowest fitting error
Recognition

• Which model to use?
• How generate hypotheses?
• How to verify these hypotheses

• Detecting planar objects
• Detecting arbitrary objects and estimate camera/object pose
Recognizing single instances

**Goal:** given a query image $I$, identify object model in the image $I$

**Model:** collection of points on a planar surface
Recognizing single instances

**Goal:** given a query image I, identify object model in the image I

**Challenges:**
- View point changes
- Illumination changes
- Features from background
Recognizing single instances

- Find matches between "model" points and "query" points
Recognizing single instances

- Find matches between “model” points and “query” points
- Using N matches to fit homographic transformation (hypothesis generation)
- If matches and selected model are correct, the fitting error is small (verification)

Verification: The hypothesis generates *high* fitting error
Recognizing single instances

- Find matches between “model” points and “query” points
- Using N matches to fit homographic transformation (hypothesis generation)
- If matches and selected model are correct, the fitting error is small (verification)

Verification: The hypothesis generates low fitting error
Recognizing single instances

- Find matches between “model” points and “query” points
- Using N matches to fit homographic transformation (hypothesis generation)
- If matches and selected model are correct, the fitting error is small (verification)

Iterate and retain hypothesis generates *lowest* fitting error
Recognizing single instances

- Find matches between “model” points and “query” points
- Using N matches to fit homographic transformation (hypothesis generation)
- If matches and selected model are correct, the fitting error is small (verification)

How to implement this?
RANSAC!
Algorithm:

1. Select random sample of minimum required size to fit model [?] = [2]
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set
Repeat 1-3 until model with the most inliers over all samples is found

Sample set = set of points in 2D

Line fitting with outliers
Line fitting with outliers

Sample set = set of points in 2D

\[ |O| = 14 \]

Algorithm:
1. Select random sample of minimum required size to fit model
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set
Repeat 1-3 until model with the most inliers over all samples is found
Line fitting with outliers

Algorithm:
1. Select random sample of minimum required size to fit model
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set
   Repeat 1-3 until model with the most inliers over all samples is found

\[ |O| = 6 \]
Recognizing single instances

**Goal:** given a query image $I$, identify object model in the image $I$

**Model:** collection of 3D points with descriptors
Recognizing single instances

**Goal:** given a query image \( I \), identify object model in the image \( I \)

**Model:** collection of 3D points with descriptors

Rothganger et al. '03 '06
Recognition

Class: toy house #3

1. Find matches between model and test image features
1. Find matches between model and test image features
2. Generate hypothesis:
   - Compute transformation $M$ from $N$ matches \((N=2; \text{ affine camera; key points with scale and rotation})\)
   - Generate hypothesis of object location and pose w.r.t. camera
Recognition

Class: toy house #3

1. Find matches between model and test image features
2. Generate hypothesis:
   • Compute transformation $M$ from $N$ matches
   • Generate hypothesis of object location and pose w.r.t. camera
3. Model verification
   • Use $M$ to project other 3D model features into test image
   • Compute residual $= D($projections, measurements$)$
Recognition

Class: toy house #3

4. Repeat steps 2 and 3 until residual doesn’t decrease anymore
5. Repeat steps 1-4 for different object instances
6. M and C corresponding to min residual return the estimated object pose and object instance
Large-scale visual search

Location recognition

Product search

Near duplicate detection

Multi-view matching

Visual link

Courtesy of Grauman and Fergus
Recent related work on large scale and efficient image search

- Kernelized Locality Sensitive Hashing for Scalable Image Search, by B. Kulis and K. Grauman, ICCV 2009
- Improving Image-based Localization by Active Correspondence Search. T. Sattler, B. Leibe, L. Kobbelt. ECCV 2012.
- City-Scale Location Recognition, G. Schindler, M. Brown, and R. Szeliski, CVPR 2007. [pdf]
Single instance object detection on a mobile device

• G. Takacs et al. "Outdoors augmented reality on mobile phone using loxel-based visual feature organization", MIR’08


Shape matching

- Match shape against database
- Retrieve relevant information

- Shape context (Belongie et al 00)
- Shape Classification Using the Inner-Distance [Ling and Jacobs 07]
Next lecture:

- Neural networks and decision trees for machine vision