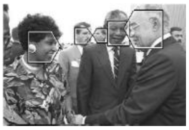
EEE422/6082 Computational Vision

Object Category Detection

Ling Shao

Some slides from Derek Hoiem

Goal: Detect all instances of objects



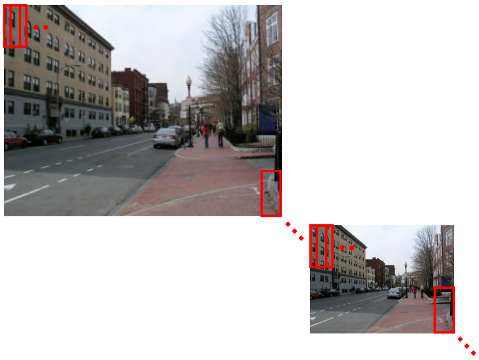
Demo: face detection

http://demo.pittpatt.com/

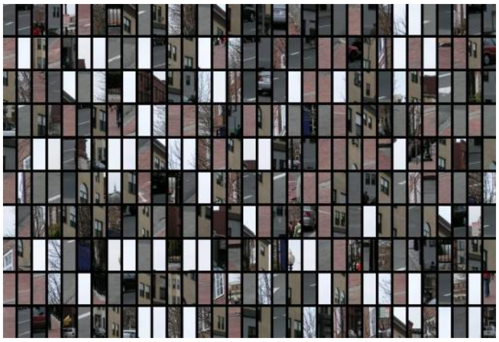
Influential Works in Detection

- Sung-Poggio (1994, 1998) : ~1450 citations
  - Basic idea of statistical template detection (I think), bootstrapping to get "face-like" negative examples, multiple whole-face prototypes (in 1994)
- Rowley-Baluja-Kanade (1996-1998) : ~2900
  - "Parts" at fixed position, non-maxima suppression, simple cascade, rotation, pretty good accuracy, fast
- Schneiderman-Kanade (1998-2000,2004) : ~1250
  - Careful feature engineering, excellent results, cascade
- Viola-Jones (2001, 2004) : ~6500
  - Haar-like features, Adaboost as feature selection, hyper-cascade, very fast, easy to implement
- Dalal-Triggs (2005) : 1025
  - Careful feature engineering, excellent results, HOG feature, online code
- Felzenszwalb-McAllester-Ramanan (2008)? 105 citations
  - Excellent template/parts-based blend

Sliding window detection



What the Detector Sees



Statistical Template

- Object model = log linear model of parts at fixed positions

+3 +2 -2 -1 -2.5 = -0.5 $\overset{?}{>}$ 7.5

Non-object

+4 +1 +0.5 +3 +0.5 = 10.5 $\overset{?}{>}$ 7.5

Object

Design challenges

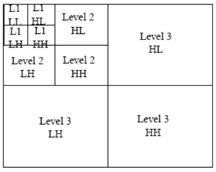
- Part design
  - How to model appearance
  - Which “parts” to include
  - How to set part likelihoods
- How to make it fast
- How to deal with different viewpoints
- Implementation details
  - Window size
  - Aspect ratio
  - Translation/scale step size
  - Non-maxima suppression

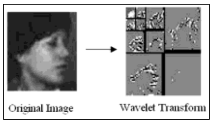
Schneiderman and Kanade

Schneiderman and Kanade. [A Statistical Method for 3D Object Detection.](#) (2000)

Parts model

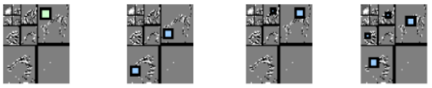
- Part = group of wavelet coefficients that are statistically dependent





Parts: groups of wavelet coefficients

- Fixed parts within/across subbands



Intra-subband | Inter-orientation | Inter-frequency | Inter-frequency/ Inter-orientation

- 17 types of “parts” that can appear at each position
- Discretize wavelet coefficient to 3 values
- E.g., part with 8 coefficients has 3^8 = 6561 values

Training

1) Create training data
   a) Get positive and negative patches
   b) Pre-process (optional), compute wavelet coefficients, discretize
   c) Compute parts values
2) Learn statistics
   a) Compute ratios of histograms by counting for positive and negative examples
   b) Reweight examples using Adaboost, recount, etc.
3) Get more negative examples (bootstrapping)

Training multiple viewpoints

Train new detector for each viewpoint.

Testing

1) Processing:
   a) Lighting correction (optional)
   b) Compute wavelet coefficients, quantize
2) Slide window over each position/scale (2 pixels, $2^{1/4}$ scale)
   a) Compute part values
   b) Lookup likelihood ratios
   c) Sum over parts
   d) Threshold
3) Use faster classifier to prune patches (cascade…more on this later)
4) Non-maximum suppression

Results: faces

Table 1. Face detection with out-of-plane rotation

γ	Detection (all faces)	Detection (profiles)	False Detections
0.0	92.7%	92.8%	700
1.5	85.5%	86.4%	91
2.5	75.2%	78.6%	12

208 images with 441 faces, 347 in profile

Results: cars

Table 3. Car detection

γ	Detections	False Detections
1.05	83%	7
1.0	86%	10
0.9	92%	71

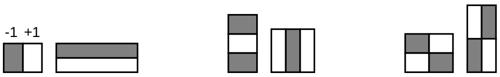
Viola and Jones

Fast detection through two mechanisms

Viola and Jones. Rapid Object Detection using a Boosted Cascade of Simple Features (2001).

Integral Images

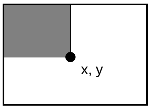
- "Haar-like features"
  - Differences of sums of intensity
  - Thousands, computed at various positions and scales within detection window



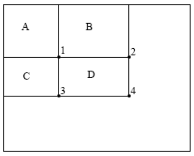
Two-rectangle features   Three-rectangle features   Etc.

Integral Images

- ii = cumsum(cumsum(Im, 1), 2)



ii(x,y) = Sum of the values in the grey region



How to compute B-A?

How to compute A+D-B-C?

Adaboost as feature selection

- Create a large pool of parts (180K)
- “Weak learner” = feature + threshold + parity

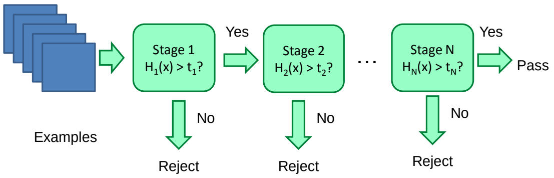
$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases}$$

- Choose weak learner that minimizes error on the weighted training set
- Reweight

Adaboost

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights,
  $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
  so that w_t is a probability distribution.
  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$.
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. Update the weights:
  $$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$
  where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:
$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
where $\alpha_t = \log \frac{1}{\beta_t}$

Cascade for Fast Detection

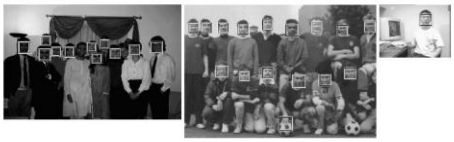


- Choose threshold for low false negative rate
- Fast classifiers early in cascade
- Slow classifiers later, but most examples don’t get there

Viola-Jones details

- 38 stages with 1, 10, 25, 50 … features
  - 6061 total used out of 180K candidates
  - 10 features evaluated on average
- Examples
  - 4916 positive examples
  - 10000 negative examples collected after each stage
- Scanning
  - Scale detector rather than image
  - Scale steps = 1.25, Translation 1.0*s to 1.5*s
- Non-max suppression: average coordinates of overlapping boxes
- Train 3 classifiers and take vote

Viola Jones Results



Detector \ False detections	10	31	50	65	78	95	167
Viola-Jones	76.1%	88.4%	91.4%	92.0%	92.1%	92.9%	93.9%
Viola-Jones (voting)	81.1%	89.7%	92.1%	93.1%	93.1%	93.2 %	93.7%
Rowley-Baluja-Kanade	83.2%	86.0%	-	-	-	89.2%	90.1%
Schneiderman-Kanade	-	-	-	94.4%	-	-	-
Roth-Yang-Ahuja	-	-	-	-	(94.8%)	-	-

MIT + CMU face dataset

Schneiderman later results

		89.7%	93.1%	94.4%	94.8%	95.7%
Schneiderman 2004	Bayesian Network *	1	8	19	36	56
	Semi-Naïve Bayes*	6	19	29	35	46
Viola-Jones 2001	[6]	31	65	--	--	--
Roth et al. 1999	[7]*	--	--	--	78	--
Schneiderman-Kanade 2000	[16]*	--	--	65	--	--

Table 2. False alarms as a function of recognition rate on the MIT-CMU Test Set for Frontal Face Detection. * indicates exclusion of the 5 images of hand-drawn faces.

Speed: frontal face detector

- Schneiderman-Kanade (2000): 5 seconds
- Viola-Jones (2001): 15 fps

Strengths and Weaknesses of Statistical Template Approach

Strengths
- Works very well for non-deformable objects: faces, cars, upright pedestrians
- Fast detection

Weaknesses
- Not so well for highly deformable objects
- Not robust to occlusion
- Requires lots of training data

SK vs. VJ

Schneiderman-Kanade
- Wavelet features
- Log linear model via boosted histogram ratios
- Bootstrap training
- Two-stage cascade
- NMS: Remove overlapping weak boxes
- Slow but very accurate

Viola-Jones
- Similar to Haar wavelets
- Log linear model via boosted stubs
- Bootstrap training
- Multistage cascade, integrated into training
- NMS: average coordinates of overlapping boxes
- Less accurate but very fast

Things to remember

- Excellent results require careful feature engineering
- Sliding window for search
- Features based on differences of intensity (gradient, wavelet, etc.)
- Boosting for feature selection (also L1-logistic regression)
- Integral images, cascade for speed
- Bootstrapping to deal with many, many negative examples