I
ROBOT
ISAAC
ASIMOV

1950
Future Vision

EYE
ROBOT
CSCI
1430

2017 MWF 1PM 368
Computer Vision

© HARPERCOLLINS / CLARE SKEATS
note: black & white
Object Detection Design challenges

• How to efficiently search for likely objects
  – Even simple models require searching hundreds of thousands of positions and scales

• Feature design and scoring
  – How should appearance be modeled?
    What features correspond to the object?

• How to deal with different viewpoints?
  – Often train different models for a few different viewpoints
Recap: Viola-Jones sliding window detector

**Fast** detection through two mechanisms
- Quickly eliminate unlikely windows
- Use features that are fast to compute

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
Features that are fast to compute

- “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

\[ ii = \text{cumsum}(\text{cumsum}(\text{im}, 1), 2) \]

\[ ii(x, y) = \text{Sum of the values in the grey region} \]

**SUM within Rectangle D is**

\[ ii(4) - ii(2) - ii(3) + ii(1) \]
Feature selection with boosting

• Create a large pool of features (180K)
• Select discriminative features that work well together

- “Weak learner” = feature + threshold + ‘polarity’

\[ h_j(x) = \begin{cases} -s_j & \text{if } f_j < \theta_j \\ s_j & \text{otherwise} \end{cases} \]

- Choose weak learner that minimizes error on the weighted training set, then reweight
1. Input the positive and negative training examples along with their labels \( \{(x_i, y_i)\} \), where \( y_i = 1 \) for positive (face) examples and \( y_i = -1 \) for negative examples.

2. Initialize all the weights to \( w_{i,1} \leftarrow \frac{1}{N} \), where \( N \) is the number of training examples. (Viola and Jones (2004) use a separate \( N_1 \) and \( N_2 \) for positive and negative examples.)

3. For each training stage \( j = 1 \ldots M \):
   
   (a) Renormalize the weights so that they sum up to 1 (divide them by their sum).
   
   (b) Select the best classifier \( h_j(x; f_j, \theta_j, s_j) \) by finding the one that minimizes the weighted classification error

   \[
   e_j = \sum_{i=0}^{N-1} w_{i,j} e_{i,j},
   \]

   \[
   e_{i,j} = 1 - \delta(y_i, h_j(x_i; f_j, \theta_j, s_j)).
   \]

   For any given \( f_j \) function, the optimal values of \( (\theta_j, s_j) \) can be found in linear time using a variant of weighted median computation (Exercise 14.2).

   (c) Compute the modified error rate \( \beta_j \) and classifier weight \( \alpha_j \),

   \[
   \beta_j = \frac{e_j}{1 - e_j} \quad \text{and} \quad \alpha_j = -\log \beta_j.
   \]

   (d) Update the weights according to the classification errors \( e_{i,j} \)

   \[
   w_{i,j+1} \leftarrow w_{i,j} \beta_j^{1 - e_{i,j}},
   \]

   i.e., downweight the training samples that were correctly classified in proportion to the overall classification error.

4. Set the final classifier to

\[
\text{sign} \left[ \sum_{j=0}^{m-1} \alpha_j h_j(x) \right].
\]
Viola Jones Results
Speed = 15 FPS (in 2001)

MIT + CMU face dataset
• Viola-Jones has a very large space of simple weak ‘edge- or pattern-like’ classifiers.
• Learn importance/spatial layout of these edges for a particular class.

• Can we use a known layout?
Object Detection

- Overview
- Viola-Jones
- Dalal-Triggs
- Deformable models
- Deep learning
Person detection with HoG’s & linear SVM’s

- Histograms of Oriented Gradients for Human Detection, Navneet Dalal, Bill Triggs, International Conference on Computer Vision & Pattern Recognition - June 2005
Statistical Template

Object model =
sum of scores of features at fixed positions

\[ +3 + 2 - 2 - 1 - 2.5 = -0.5 > 7.5 \]
Non-object

\[ +4 + 1 + 0.5 + 3 + 0.5 = 10.5 > 7.5 \]
Object
Example: Dalal-Triggs pedestrian detector

1. Extract fixed-sized (64x128 pixel) window at each position and scale
2. Compute HOG (histogram of gradient) features within each window
3. Score the window with a linear SVM classifier
4. Perform non-maxima suppression to remove overlapping detections with lower scores
Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05
• Tested with
  – RGB
  – LAB
  – Grayscale

• Gamma Normalization and Compression
  – Square root
  – Log

Slightly better performance vs. grayscale
Very slightly better performance vs. no adjustment
Outperforms

Centered:
-1 0 1

Uncentered:
-1 1

Cubic-corrected:
1 -8 0 8 -1

Diagonal:
0 1
-1 0

Sobel:
-1 0 1
-2 0 2
-1 0 1
- Histogram of Oriented Gradients

Orientation: 9 bins (for unsigned angles 0 - 180)

- Votes weighted by magnitude
- Bilinear interpolation between cells

Histograms in k x k pixel cells

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05
Normalize with respect to surrounding cells

\[
f = \frac{v}{\sqrt{\|v\|^2_2 + e^2}}
\]

\(e\) is a small constant
Histograms of Oriented Gradients for Human Detection

Input image → Normalize gamma & colour → Compute gradients → Weighted vote into spatial & orientation cells → Contrast normalize over overlapping spatial blocks → Collect HOG’s over detection window → Linear SVM → Person / non-person classification

\[ X = \]

# features = 15 \times 7 \times 9 \times 4 = 3780

# cells

# normalizations by neighboring cells

# orientations
Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05
0.16 = w^T x - b

\text{sign}(0.16) = 1

\implies \text{pedestrian}
Pedestrian detection with HOG

- Learn a pedestrian template using a support vector machine
- At test time, convolve feature map with template
- Find local maxima of response
- For multi-scale detection, repeat over multiple levels of a HOG pyramid

Something to think about...

• Sliding window detectors work
  – *very well* for faces
  – *fairly well* for cars and pedestrians
  – *badly* for cats and dogs

• Why are some classes easier than others?
Strengths/Weaknesses of Statistical Template Approach

Strengths

• Works very well for non-deformable objects with canonical orientations: faces, cars, pedestrians
• Fast detection

Weaknesses

• Not so well for highly deformable objects or “stuff”
• Not robust to occlusion
• Requires lots of training data
Tricks of the trade

• Details in feature computation really matter
  – E.g., normalization in Dalal-Triggs improves detection rate by 27% at fixed false positive rate

• Template size
  – Typical choice is size of smallest expected detectable object

• “Jittering” or “augmenting” to create synthetic positive examples
  – Create slightly rotated, translated, scaled, mirrored versions as extra positive examples.

• Bootstrapping to get hard negative examples
  1. Randomly sample negative examples
  2. Train detector
  3. Sample negative examples that score > -1
  4. Repeat until all high-scoring negative examples fit in memory
Things to remember

• Sliding window for search

• Features based on differences of intensity (gradient, wavelet, etc.)
  – Excellent results = careful feature design

• Boosting for feature selection

• Integral images, cascade for speed

• Bootstrapping to deal with many, many negative examples
Project 5

• Train Dalal-Triggs model for faces
• Classify examples

• We need some test photographs...