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
Integral Images

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

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

\[ \text{SUM within Rectangle D is} \]
\[ ii(4) - ii(2) - ii(3) + ii(1) \]
Feature selection with Adaboost

• Create a large pool of features (180K)
• Select features that are discriminative and work well together
  – “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
Viola Jones Results
Speed = 15 FPS (in 2001)

<table>
<thead>
<tr>
<th>Detector</th>
<th>10</th>
<th>31</th>
<th>50</th>
<th>65</th>
<th>78</th>
<th>95</th>
<th>167</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viola-Jones</td>
<td>76.1%</td>
<td>88.4%</td>
<td>91.4%</td>
<td>92.0%</td>
<td>92.1%</td>
<td>92.9%</td>
<td>93.9%</td>
</tr>
<tr>
<td>Viola-Jones (voting)</td>
<td>81.1%</td>
<td>89.7%</td>
<td>92.1%</td>
<td>93.1%</td>
<td>93.1%</td>
<td>93.2%</td>
<td>93.7%</td>
</tr>
<tr>
<td>Rowley-Baluja-Kanade</td>
<td>83.2%</td>
<td>86.0%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>89.2%</td>
<td>90.1%</td>
</tr>
<tr>
<td>Schneiderman-Kanade</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>94.4%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Roth-Yang-Ahuja</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>(94.8%)</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

MIT + CMU face dataset
Object Detection

• Overview
• Viola-Jones
• Dalal-Triggs
• Deformable models
• Deep learning
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
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

-1 0 1
centered

-1 1
uncentered

1 -8 0 8 -1
cubic-corrected

0 1
diagonal

-1 0 1
-2 0 2
-1 0 1
Sobel

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05
• Histogram of gradient orientations

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

Histograms in k x k pixel cells

– Votes weighted by magnitude
– Bilinear interpolation between cells
Normalize with respect to surrounding cells

\[ L^2 - \text{norm}: \ v \rightarrow \frac{v}{\sqrt{\|v\|_2^2 + \epsilon^2}} \]
X =

Original Formulation

$\# \text{orientations}$

$\# \text{features} = 15 \times 7 \times 9 \times 4 = 3780$

$\# \text{cells}$

$\# \text{normalizations by neighboring cells}$
$0.16 = w^T x - b$

$\text{sign}(0.16) = 1$

$\implies$ pedestrian
Pedestrian detection with HOG

- Train a pedestrian template using a linear 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 and 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 detectable object

• “Jittering” 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 require careful feature design

• Boosting for feature selection

• Integral images, cascade for speed

• Bootstrapping to deal with many, many negative examples

---

Examples

Stage 1

\[ H_1(x) > t_1? \]

Reject

Stage 2

\[ H_2(x) > t_2? \]

Yes

Stage N

\[ H_N(x) > t_N? \]

Yes

Pass

Reject

Reject

Reject

...
<table>
<thead>
<tr>
<th>Root filter</th>
<th>Part filters</th>
<th>Deformation weights</th>
</tr>
</thead>
</table>

Car model

Component 1

Component 2
Person model
Bottle model
More detections

horse

sofa

bottle
The PASCAL Visual Object Classes Challenge 2009 (VOC2009)

- Twenty object categories (aeroplane to TV/monitor)

- Three challenges:
  - Classification challenge (is there an X in this image?)
  - Detection challenge (draw a box around every X)
  - Segmentation challenge

Slides from Noah Snavely
Dataset: Collection

- Images downloaded from **flickr**
  - 500,000 images downloaded and random subset selected for annotation
Dataset: Annotation

- Complete annotation of all objects
- Annotated over web with written guidelines
  - High quality (?)
Classification Challenge

- Predict whether at least one object of a given class is present in an image

is there a cat?
Evaluation

- **Average Precision [TREC]** averages precision over the entire range of recall
  - Curve interpolated to reduce influence of “outliers”

- A good score requires both high recall and high precision
- Application-independent
- Penalizes methods giving high precision but low recall
Participation

- 48 Methods, 20 Groups
Results: AP by Method and Class

<table>
<thead>
<tr>
<th>Method</th>
<th>AP</th>
<th>bicycle</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>dining</th>
<th>table</th>
<th>dog</th>
<th>horse</th>
<th>bike</th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv/monitor</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVC_FLAT</td>
<td>85.3</td>
<td>57.8</td>
<td>66.0</td>
<td>66.1</td>
<td>36.2</td>
<td>70.6</td>
<td>60.6</td>
<td>63.5</td>
<td>55.1</td>
<td>44.6</td>
<td>53.4</td>
<td>49.1</td>
<td>64.4</td>
<td>66.8</td>
<td>84.8</td>
<td>37.4</td>
<td>44.1</td>
<td>47.9</td>
<td>81.9</td>
<td>67.5</td>
<td></td>
</tr>
<tr>
<td>CVC_FLAT-HOG-ESS</td>
<td>86.3</td>
<td>60.7</td>
<td>66.4</td>
<td>65.3</td>
<td>41.0</td>
<td>71.7</td>
<td>64.7</td>
<td>63.9</td>
<td>55.5</td>
<td>40.1</td>
<td>51.3</td>
<td>45.9</td>
<td>65.2</td>
<td>68.9</td>
<td>85.0</td>
<td>40.8</td>
<td>49.0</td>
<td>49.1</td>
<td>81.8</td>
<td>68.6</td>
<td></td>
</tr>
<tr>
<td>CVC_PLUS</td>
<td>86.6</td>
<td>58.4</td>
<td>66.7</td>
<td>67.3</td>
<td>34.8</td>
<td>70.4</td>
<td>60.0</td>
<td>64.2</td>
<td>52.5</td>
<td>43.0</td>
<td>50.8</td>
<td>46.5</td>
<td>64.1</td>
<td>66.8</td>
<td>84.4</td>
<td>37.5</td>
<td>45.1</td>
<td>45.4</td>
<td>82.1</td>
<td>67.0</td>
<td></td>
</tr>
<tr>
<td>FIRSTNIKON_AVGSRKDA</td>
<td>83.3</td>
<td>59.3</td>
<td>62.7</td>
<td>65.3</td>
<td>30.2</td>
<td>71.6</td>
<td>58.2</td>
<td>62.2</td>
<td>54.3</td>
<td>40.7</td>
<td>49.2</td>
<td>50.0</td>
<td>66.6</td>
<td>62.9</td>
<td>83.3</td>
<td>34.2</td>
<td>48.2</td>
<td>46.1</td>
<td>83.4</td>
<td>65.5</td>
<td></td>
</tr>
<tr>
<td>FIRSTNIKON_AVGsvm</td>
<td>83.8</td>
<td>58.2</td>
<td>62.6</td>
<td>65.2</td>
<td>32.0</td>
<td>69.8</td>
<td>57.7</td>
<td>61.1</td>
<td>54.5</td>
<td>44.0</td>
<td>50.3</td>
<td>49.6</td>
<td>64.6</td>
<td>61.7</td>
<td>83.2</td>
<td>33.4</td>
<td>46.5</td>
<td>48.0</td>
<td>81.6</td>
<td>65.3</td>
<td></td>
</tr>
<tr>
<td>FIRSTNIKON_BOOSTSRKDA</td>
<td>83.0</td>
<td>59.2</td>
<td>61.4</td>
<td>64.6</td>
<td>33.2</td>
<td>71.1</td>
<td>57.5</td>
<td>61.0</td>
<td>54.8</td>
<td>40.7</td>
<td>48.3</td>
<td>50.0</td>
<td>65.5</td>
<td>63.4</td>
<td>82.8</td>
<td>32.8</td>
<td>47.0</td>
<td>47.1</td>
<td>83.3</td>
<td>64.6</td>
<td></td>
</tr>
<tr>
<td>FIRSTNIKON_BOOSTsvm</td>
<td>83.5</td>
<td>56.8</td>
<td>61.8</td>
<td>65.5</td>
<td>33.2</td>
<td>69.7</td>
<td>57.3</td>
<td>60.5</td>
<td>54.6</td>
<td>43.1</td>
<td>48.3</td>
<td>50.3</td>
<td>64.3</td>
<td>62.4</td>
<td>82.3</td>
<td>32.9</td>
<td>46.9</td>
<td>48.4</td>
<td>82.0</td>
<td>64.2</td>
<td></td>
</tr>
<tr>
<td>LEAR_CHI-SVM-MULT-LOC</td>
<td>79.5</td>
<td>55.5</td>
<td>54.5</td>
<td>63.9</td>
<td>43.7</td>
<td>70.3</td>
<td>66.4</td>
<td>56.5</td>
<td>54.4</td>
<td>38.8</td>
<td>44.1</td>
<td>46.2</td>
<td>58.5</td>
<td>64.2</td>
<td>82.2</td>
<td>39.1</td>
<td>41.3</td>
<td>39.8</td>
<td>73.6</td>
<td>66.2</td>
<td></td>
</tr>
<tr>
<td>NECUIUC_CDCV</td>
<td>88.1</td>
<td>68.0</td>
<td>68.0</td>
<td>72.5</td>
<td>41.0</td>
<td>78.9</td>
<td>70.4</td>
<td>70.4</td>
<td>58.1</td>
<td>53.4</td>
<td>55.7</td>
<td>59.3</td>
<td>73.1</td>
<td>71.3</td>
<td>84.5</td>
<td>32.3</td>
<td>53.3</td>
<td>56.7</td>
<td>86.0</td>
<td>66.8</td>
<td></td>
</tr>
<tr>
<td>NECUIUC_CLS-DTCT</td>
<td>88.0</td>
<td>68.6</td>
<td>67.9</td>
<td>72.9</td>
<td>44.2</td>
<td>79.5</td>
<td>72.5</td>
<td>70.8</td>
<td>59.5</td>
<td>53.6</td>
<td>57.5</td>
<td>59.0</td>
<td>72.6</td>
<td>72.3</td>
<td>85.3</td>
<td>36.6</td>
<td>56.9</td>
<td>57.9</td>
<td>85.9</td>
<td>68.0</td>
<td></td>
</tr>
<tr>
<td>NECUIUC_LL-CDCV</td>
<td>87.1</td>
<td>67.4</td>
<td>65.8</td>
<td>72.3</td>
<td>40.9</td>
<td>78.3</td>
<td>69.7</td>
<td>69.7</td>
<td>58.5</td>
<td>50.1</td>
<td>55.1</td>
<td>56.3</td>
<td>71.8</td>
<td>70.8</td>
<td>84.1</td>
<td>31.4</td>
<td>51.5</td>
<td>55.1</td>
<td>84.7</td>
<td>65.2</td>
<td></td>
</tr>
<tr>
<td>NECUIUC_LN-CDCV</td>
<td>87.7</td>
<td>67.8</td>
<td>68.1</td>
<td>71.1</td>
<td>39.1</td>
<td>78.5</td>
<td>70.6</td>
<td>70.7</td>
<td>57.4</td>
<td>51.7</td>
<td>53.3</td>
<td>59.2</td>
<td>71.6</td>
<td>70.6</td>
<td>84.0</td>
<td>30.9</td>
<td>51.7</td>
<td>55.9</td>
<td>85.9</td>
<td>66.7</td>
<td></td>
</tr>
<tr>
<td>UVASURREY_BASELINE</td>
<td>84.1</td>
<td>59.2</td>
<td>62.7</td>
<td>65.4</td>
<td>35.7</td>
<td>70.6</td>
<td>59.8</td>
<td>61.3</td>
<td>56.7</td>
<td>45.3</td>
<td>52.4</td>
<td>50.6</td>
<td>66.1</td>
<td>66.6</td>
<td>83.7</td>
<td>34.8</td>
<td>47.2</td>
<td>47.7</td>
<td>80.8</td>
<td>65.9</td>
<td></td>
</tr>
<tr>
<td>UVASURREY_MKFDA+BOW</td>
<td>84.7</td>
<td>63.9</td>
<td>66.1</td>
<td>67.3</td>
<td>37.9</td>
<td>74.1</td>
<td>63.2</td>
<td>64.0</td>
<td>57.1</td>
<td>46.2</td>
<td>54.7</td>
<td>53.5</td>
<td>68.1</td>
<td>70.6</td>
<td>85.2</td>
<td>38.5</td>
<td>47.2</td>
<td>49.3</td>
<td>83.2</td>
<td>68.1</td>
<td></td>
</tr>
<tr>
<td>UVASURREY_TUNCOLORKERNELSEL</td>
<td>85.0</td>
<td>62.8</td>
<td>65.1</td>
<td>66.8</td>
<td>37.6</td>
<td>73.5</td>
<td>62.1</td>
<td>62.0</td>
<td>57.4</td>
<td>45.1</td>
<td>54.5</td>
<td>52.5</td>
<td>67.7</td>
<td>69.8</td>
<td>84.8</td>
<td>39.1</td>
<td>46.8</td>
<td>49.9</td>
<td>82.9</td>
<td>68.1</td>
<td></td>
</tr>
<tr>
<td>UVASURREY_TUNCOLORSPECKDA</td>
<td>84.6</td>
<td>62.4</td>
<td>65.6</td>
<td>67.2</td>
<td>39.4</td>
<td>74.0</td>
<td>63.4</td>
<td>62.8</td>
<td>56.7</td>
<td>43.8</td>
<td>54.7</td>
<td>52.7</td>
<td>67.3</td>
<td>70.6</td>
<td>85.0</td>
<td>38.8</td>
<td>46.9</td>
<td>50.0</td>
<td>82.2</td>
<td>66.2</td>
<td></td>
</tr>
</tbody>
</table>

- Only methods in 1st, 2nd or 3rd place by group shown
- Groups: CVC, FIRST/Nikon, NEC/UIUC, UVA/Surrey
- Max AP: 88.1% (aeroplane) ... 40.8% (potted plant)
Precision/Recall: Aeroplane (All)
Precision/Recall: Potted plant (Top 10 by AP)
Ranked Images: Aeroplane

- Class images: Highest ranked
Ranked Images: Chair

- Class images: Highest ranked
Detection Challenge

- Predict the bounding boxes of all objects of a given class in an image (if any)
Evaluating Bounding Boxes

- Area of Overlap (AO) Measure

\[ AO(B_{gt}, B_{p}) = \frac{|B_{gt} \cap B_{p}|}{|B_{gt} \cup B_{p}|} \]

- Need to define a threshold \( t \) such that \( AO(B_{gt}, B_{p}) \) implies a correct detection: 50\%
AP by Class

Chance essentially 0
Precision/Recall - Aeroplane
Precision/Recall - Car
True Positives - Person

UoCTTI_LSVM-MDPM

MIZZOU_DEF-HOG-LBP

NECUIUC_CLS-DTCT
False Positives - Person

UoCTTI_L SVM-MDPM

MIZZOU_DEF-HOG-LBP

NECU1UC_CLS-D TCT
True Positives - Bicycle

UoCTTI_LSVM-MDPM

OXFORD_MKL

NECUIUC_CLS-DTCT
False Positives - Bicycle

UoCTTI_LSVM-MDPM

OXFORD_MKL

NECUIUC_CLS-DTCT