What is Object Detection?

Formally: Object detection is deciding if a new image belongs to the set of images of an object.

Conservative assumption: Increasing object variability increases difficulty of detection problem.
Sources of Variability

- Image plane variation (rotation, translation, scale)
- Object pose (3D rotation, distance from camera)
- Lighting and surface appearance / texture
- Background variation
- Shape variation (within class: cars or chairs)
- Shape variation (within object: articulated motion)
Building an Object Detector

• How to **partition** problem?
  • Separating images by pose (profile / frontal face)
  • Sub-features of object (eyes, nose, mouth)

• How to do **classification**?
  • Preprocessing of images
  • Type of classifier
  • Training procedure

• How to **merge** results?
  • Graph matching
  • Statistical methods
Architecture of Frontal Face Detector

- Extract 20 x 20 pixel windows from the image
- Preprocess windows to improve contrast / lighting
- Apply (multiple) neural networks for classification
- Arbitrate among networks
Extracting 20 x 20 Windows

To make detection simpler, just detect faces centered in, and filling, 20 x 20 pixel windows.
Preprocessing Windows

Lighting and contrast may be poor in the images.

Original window:

Best fit plane:

Original minus best fit plane:

Apply histogram equalization:
Positive Training Examples
Randomizing Positive Examples
Selecting Negative Examples

Selecting a representative sample of non-faces is hard.

Active Learning

1. Train network on training set.
2. Present an image which contains no faces.
3. See where it makes mistakes.
4. Add mistakes to training set as negative examples.
5. Repeat.
Negative Examples
Clean-Up Heuristics

AND of networks

AND + merge detections

AND + merging + remove overlaps
Results: Digitized TV Images
Results: Sitcom Casts
Results: Musicians
Results: Movie Stars
Results: Random Images
Results: Class Picture
## Accuracy

Sung & Poggio: 23 images, 155 faces, 9678084 windows

<table>
<thead>
<tr>
<th>System</th>
<th>Detect Rate %</th>
<th>False Detects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single network</td>
<td>92.9</td>
<td>353</td>
</tr>
<tr>
<td>Single network + heuristics</td>
<td>92.3</td>
<td>126</td>
</tr>
<tr>
<td>Two networks (version 1)</td>
<td>78.1</td>
<td>3</td>
</tr>
<tr>
<td>Two networks (version 2)</td>
<td>87.1</td>
<td>15</td>
</tr>
<tr>
<td>Two networks (version 3)</td>
<td>92.9</td>
<td>64</td>
</tr>
<tr>
<td>MIT: PCA/Clustering/MLP</td>
<td>76.8</td>
<td>5</td>
</tr>
<tr>
<td>MIT: PCA/Clustering/Perceptron</td>
<td>81.9</td>
<td>13</td>
</tr>
<tr>
<td>Fast Version</td>
<td>72.9</td>
<td>3</td>
</tr>
</tbody>
</table>

Version 1: AND network outputs, then apply a threshold and overlap elimination
Version 2: Apply heuristics to networks separately, then AND the results
Version 3: Apply heuristics to networks separately, OR the results, then remove overlaps
Variations on Face Detection

- Speed improvements
- Sub-features of face: Eye detection
- Profiles and other views
Speed Improvements

Processing time for a 320x240 image:
5.5 minutes on an SGI Indigo 2.

Where is the bottleneck?
Must extract 20x20 pixel window from every pixel position and scale.

Solution from license plate detector, Umezaki [1995]:
Do not examine each pixel location.
Speed Improvements

Use the same training procedure, different data:

- Larger input window: 30x30 pixels
- Positive examples no longer centered:

- Detector moves in steps of 10 pixels over image
- Single output indicates presence of a face
Accuracy of Large Window Detector

Many more false detections than Small Window detector

Small Window Version

Large Window Version
Improving Accuracy

Treat Large Window detections as candidates.
Verify candidates with Small Window detector.
Speedup

320x240 image, with Large Window detector:

- 9 seconds on an SGI Indigo 2.
- Faster by a factor of 35.

Further Speedups

- Motion 3 - 5 seconds
- Skin color detection, Yang & Waibel [1995] 1 - 2 seconds
- Temporal coherence / tracking 0.2 seconds
Sub-features of Face: Eyes
Same training procedure, new data (25x15 windows):

More false detections than for faces:

Eye detector alone
With face location
Partitioning / Merging Views of Faces

Architecture suggested by Baluja [1996]:

View Recognizer

- Left Profile
- Half Left
- Frontal Face
- Half Right
- Right Profile

Or

Output

View Recognizer accuracy of 99% is easily achieved.
Pose Invariant Face Detection