Face Detection

February 26, 2007

Lecture Plan

• HW3: Due tonight
• Project Updates
• Face Detection: Neural Networks Approach
• Face Detection: Cascades
• OpenCV Demo

Face Detection v.s. Face Recognition


“Rotation Invariant Neural Network-Based Face Detection,”
Detection Network

Testing on Images with no Faces

- All detections are automatically false positives
- They are added as negative examples in the training database

Results

<table>
<thead>
<tr>
<th>System</th>
<th>Upright Test Set</th>
<th>Rotated Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Detect %</td>
<td># False</td>
</tr>
<tr>
<td>Network 1</td>
<td>89.6%</td>
<td>4855</td>
</tr>
<tr>
<td>Network 2</td>
<td>85.5%</td>
<td>4111</td>
</tr>
<tr>
<td>Net 1 → Postproc</td>
<td>85.7%</td>
<td>2024</td>
</tr>
<tr>
<td>Net 2 → Postproc</td>
<td>84.1%</td>
<td>1728</td>
</tr>
<tr>
<td>Postproc → AND</td>
<td>81.6%</td>
<td>293</td>
</tr>
</tbody>
</table>

Face Detection Movies
Web Demo of Face Detection

http://demo.pittpatt.com/

Face Detection Using Cascades


Viola/Jones Face Detector

• Technical advantages:
  – Uses lots of very simple box features, enabling an efficient image representation
  – Scales features rather than source image
  – Cascaded classifier is very fast on non-faces
• Practical benefits:
  – Very fast, compact footprint
  – You don’t have to implement it!
    (should be in latest version of OpenCV)

Classical Face Detection

This next set of slides is from:

Robust Real-time Object Detection
by
Paul Viola and Michael Jones
ICCV 2001 Workshop on Statistical and Computation Theories of Vision
Presentation by Gyozo Gidofalvi
Computer Science and Engineering Department
University of California, San Diego
gyozo@cs.ucsd.edu
October 25, 2001
Object detection task

- Object detection framework: Given a set of images find regions in these images which contain instances of a certain kind of object.
- Task: Develop an algorithm to learn an fast and accurate method for object detection.

To capture ad-hoc domain knowledge classifiers for images do not operate on raw grayscale pixel values but rather on values obtained from applying simple filters to the pixels.

Definition of simple features for object detection

- 3 rectangular features types:
  - two-rectangle feature type (horizontal/vertical)
  - three-rectangle feature type
  - four-rectangle feature type

Using a 24x24 pixel base detection window, with all the possible combination of horizontal and vertical location and scale of these feature types the full set of features has 49,396 features.

The motivation behind using rectangular features, as opposed to more expressive steerable filters is due to their extreme computational efficiency.

Rapid evaluation of rectangular features

Using the integral image representation one can compute the value of any rectangular sum in constant time.

For example the integral sum inside rectangle D we can compute as:

\[ \text{Area(D)} = \text{Area(4,1)} - \text{Area(2,1)} - \text{Area(4,3)} + \text{Area(2,3)} \]

As a result two-, three-, and four-rectangular features can be computed with 6, 8 and 9 array references respectively.

Integral image

Definition: The integral image at location \((x,y)\) is the sum of the pixel values above and to the left of \((x,y)\), inclusive.

Using the following two recurrences, where \(i(x,y)\) is the pixel value of original image at the given location and \(s(x,y)\) is the cumulative column sum, we can calculate the integral image representation of the image in a single pass.

\[
\begin{align*}
  s(x,y) &= s(x,y-1) + i(x,y) \\
  ii(x,y) &= ii(x-1,y) + s(x,y)
\end{align*}
\]

Challenges for learning a classification function

- Given a feature set and labeled training set of images one can apply number of machine learning techniques.
- Recall however, that there is 45,396 features associated with each image sub-window, hence the computation of all features is computationally prohibitive.
- Hypothesis: A combination of only a small number of these features can yield an effective classifier.
- Challenge: Find these discriminant features.
Performance of 200 feature face detector

The ROC curve of the constructed classifier indicates that a reasonable detection rate of 0.95 can be achieved while maintaining an extremely low false positive rate of approximately $10^{-4}$.

- First features selected by AdaBoost are meaningful and have high discriminative power.
- By varying the threshold of the final classifier one can construct a two-feature classifier which has a detection rate of 1 and a false positive rate of 0.4.

Processing in / training of the Attentional Cascade

Processing is essentially identical to the processing performed by a degenerate decision tree, namely only a positive result from a previous classifier triggers the evaluation of the subsequent classifier.

Training is also much like the training of a decision tree, namely subsequent classifiers are trained only on examples which pass through all the previous classifiers. Hence the task faced by classifiers further down the cascade is more difficult.

To achieve efficient cascade for a given false positive rate $F$ and detection rate $D$ we would like to minimize the expected number of features evaluated $N$:

$$N = n_1 \sum_i (n_i \prod_i p_i)$$

Since this optimization is extremely difficult the usual framework is to choose a minimal acceptable false positive and detection rate per layer.

Experiments cont. (structure of the detector cascade)

- The final detector had 32 layers and 4297 features total.
- Speed of the detector ~ total number of features evaluated
- On the MIT-CMU test set the average number of features evaluated is 8 (out of 4297).
- The processing time of a 384 by 288 pixel image on a conventional personal computer about 0.067 seconds.
- On the MIT-CMU test set the average number of features evaluated is 8 (out of 4297).
- Processing time should linearly scale with image size, hence processing of a 3.1 mega pixel images taken from a digital camera should approximately take 2 seconds.

Experiments (dataset for training)

- 4916 positive training examples were hand picked, aligned, normalized, and scaled to a base resolution of 24x24.
- 10,000 negative examples were selected by randomly picking sub-windows from 9500 images which did not contain faces.

Results

Testing of the final face detector was performed using the MIT-CMU frontal face test which consists of:

- 130 images
- 505 labeled frontal faces

Results in the table compare the performance of the detector to best face detectors known.

- Rowley at al.: use a combination of two neural networks (simple network for prescreening larger regions, complex network for detection of faces).
- Schneiderman at al.: use a set of models to capture the variation in facial appearance; each model describes the statistical behavior of a group of wavelet coefficients.
Results cont.

Conclusion

• The paper presents general object detection method which is illustrated on the face detection task.
• Using the integral image representation and simple rectangular features eliminate the need of expensive calculation of multi-scale image pyramid.
• Simple modification to AdaBoost gives a general technique for efficient feature selection.
• A general technique for constructing a cascade of homogeneous classifiers is presented, which can reject most of the negative examples at early stages of processing thereby significantly reducing computation time.
• A face detector using these techniques is presented which is comparable in classification performance to, and orders of magnitude faster than the best detectors known today.

Live OpenCV Demo of Face Detection Using Cascades

THE END