FaceTracer:
A Search Engine for Large Collections of Images with Faces

Authors:
Neeraj Kumar,
Peter Belhumeur, Shree Nayar
Columbia University

Presented by: Girish Malkarnenkar
19th October 2012
CS395T Visual Recognition
Motivation

• Current search engines use text annotations to find images based on facial appearance.

• **Problems** with this approach:
  1. Manual labeling is time consuming
  2. Textual annotations can be misleading/incorrect
  3. Annotated images are only a small subset of all the images
Google Images then...
Their method then...
Their method now...
Google Images now...
PROBLEM STATEMENT:

1. Goal: A search engine based on both Facial and Image appearance

2. Since there are billions of images and hundreds of possible attributes, and we can only hope to get a few thousands of manual labels, the labeling of images needs to be done automatically in a scalable manner
Database Creation: Downloading images

Celebrity names, Professions, Events etc

Keyword Searches → Image Downloader → Image Database → Face and Feature Detector → Resolution and Pose Filter

Random Downloads → Attribute Database → Attribute Classifiers → Face Database → Affine Transformation

Randomly downloaded to permit sampling from a general distribution...

Image from: ECCV 2008 paper, Logos from: link1, link2, link3
Database Creation: Face detection

Detected Face + Pose angles + Locations of 6 points (corners of eyes + mouth)

Image from: ECCV 2008 paper, OKAO logo: link
Database Creation: Filter/Transformation

Affine transformation to canonical frontal pose using least squares on the 6 points w.r.t a template

Filter detected faces by pose (+/- 10 degrees from front/center)

# Image Database Statistics

<table>
<thead>
<tr>
<th>Image Source</th>
<th># Images</th>
<th># Faces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Randomly Downloaded</td>
<td>4,248,194</td>
<td>2,124,472</td>
</tr>
<tr>
<td>Celebrities</td>
<td>105,568</td>
<td>109,748</td>
</tr>
<tr>
<td>Person Names</td>
<td>19,492</td>
<td>12,806</td>
</tr>
<tr>
<td>Face-Related Words</td>
<td>13,212</td>
<td>14,424</td>
</tr>
<tr>
<td>Event-Related Words</td>
<td>1,429</td>
<td>1,335</td>
</tr>
<tr>
<td>Professions</td>
<td>115,808</td>
<td>79,992</td>
</tr>
<tr>
<td>Series</td>
<td>7,551</td>
<td>8,585</td>
</tr>
<tr>
<td>Camera Defaults</td>
<td>2,153</td>
<td>879</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>10,855</td>
<td>16,201</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>4,539,886</strong></td>
<td><strong>2,373,533</strong></td>
</tr>
</tbody>
</table>

## Database Size Comparison

<table>
<thead>
<tr>
<th>Database</th>
<th># Face Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIT+CMU</td>
<td>130</td>
</tr>
<tr>
<td>Yale A</td>
<td>165</td>
</tr>
<tr>
<td>Yale B</td>
<td>5,760</td>
</tr>
<tr>
<td>FERET</td>
<td>14,051</td>
</tr>
<tr>
<td>CMU PIE</td>
<td>41,368</td>
</tr>
<tr>
<td>FRGC v2.0</td>
<td>50,000</td>
</tr>
<tr>
<td>Proposed</td>
<td>2,373,533</td>
</tr>
</tbody>
</table>

Total Number of Faces

Total Number of Faces

Total Number of Faces

Total Number of Faces

MIT+CMU
Yale A
Yale B
FERET
CMU PIE
FRGC v2.0

Manual labeling of attributes...

So at this stage, we **have ~3.1 million images** (at the time of publication in 2008) and we need to train attribute classifier on them for 10 attributes

It is infeasible to manually label all the 3.1M images

**BUT**

we do need some labeled images for automatically labeling the remaining, so **we manually create ~17,000 attribute labeled images**

Image from: ECCV 2008 paper
### Labeled Attribute Statistics

<table>
<thead>
<tr>
<th>Attribute</th>
<th># Labeled</th>
<th>Attribute</th>
<th># Labeled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td>1,954</td>
<td>Mustache</td>
<td>1,947</td>
</tr>
<tr>
<td>Age</td>
<td>3,301</td>
<td>Smiling</td>
<td>1,571</td>
</tr>
<tr>
<td>Race</td>
<td>1,309</td>
<td>Blurry</td>
<td>1,763</td>
</tr>
<tr>
<td>Hair Color</td>
<td>1,033</td>
<td>Lighting</td>
<td>633</td>
</tr>
<tr>
<td>Eye Wear</td>
<td>2,360</td>
<td>Environment</td>
<td>1,583</td>
</tr>
</tbody>
</table>

**Total Number of Labels: 17,454**

And this is where the fun starts...

- Goal: Given the 17k attribute labels we now need to train attribute classifiers for all 10 attributes to automatically label the remaining images...
Where to extract features from?

Face divided into 10 functional regions...

Image from: ECCV 2008 paper
## Feature Types

<table>
<thead>
<tr>
<th>Pixel Value Type</th>
<th>Normalizations</th>
<th>Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB (r)</td>
<td>None (n)</td>
<td>None (n)</td>
</tr>
<tr>
<td>HSV (h)</td>
<td>Mean-Norm (m)</td>
<td>Histogram (h)</td>
</tr>
<tr>
<td>Image Intensity (i)</td>
<td>Energy-Norm (e)</td>
<td>Statistics (s)</td>
</tr>
<tr>
<td>Edge Magnitude (m)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edge Orientation (o)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

# Feature Types

<table>
<thead>
<tr>
<th>Pixel Value Type</th>
<th>Normalizations</th>
<th>Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB (r)</td>
<td>None (n)</td>
<td>None (n)</td>
</tr>
<tr>
<td>HSV (h)</td>
<td>Mean-Norm (m)</td>
<td>Histogram (h)</td>
</tr>
<tr>
<td>Image Intensity (i)</td>
<td>Energy-Norm (e)</td>
<td>Statistics (s)</td>
</tr>
<tr>
<td>Edge Magnitude (m)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edge Orientation (o)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RGB, Mean Norm., No Aggreg. (r.m.n)

<table>
<thead>
<tr>
<th>Pixel Value Type</th>
<th>Normalizations</th>
<th>Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB (r)</td>
<td>None (n)</td>
<td>None (n)</td>
</tr>
<tr>
<td>HSV (h)</td>
<td>Mean-Norm (m)</td>
<td>Histogram (h)</td>
</tr>
<tr>
<td>Image Intensity (i)</td>
<td>Energy-Norm (e)</td>
<td>Statistics (s)</td>
</tr>
<tr>
<td>Edge Magnitude (m)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edge Orientation (o)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Edge Orientations, No Norm, Histogram (o.n.h)**

Classifier architecture...

• Recent state of the art results in classification have mainly been achieved with SVMs
• The problem with SVMs is that irrelevant features can confuse/over-train the classifier...
• E.g. It might not make sense to use all facial pixels for training a classifier for just “is smiling”
• Given the large set of types of features/regions, we need a good way of selecting an optimal combination of features for each attribute...
• Enter Adaboost...
Quick Review of Boosting...

Boosting illustration

Boosting illustration

Weights Increased

Boosting illustration

Final classifier is a combination of weak classifiers

Combining Boosting with SVMs...

• The idea is to construct a “local” SVM for every possible combination of region, feature types and SVM parameters (LibSVM)

• And then to use Adaboost to create an optimal classifier using a linear combination of these local SVMs

• The usual Adaboost algorithm is modified so that no retraining is needed at the beginning of each round (since these SVMs are either powerful/useless classifiers depending on the relevance of the features used)
Discussion...

• Boosting is meant to turn weak learners into strong learners. Does using boosting in this scenario where you have pre-trained SVMs make sense? Wouldn’t using some feature selection approach be better?

• Performance degradation in boosting, (Wickramaratna, J. and Holden, S. and Buxton, B., Multiple Classifier Systems, 2001) shows that boosting strong learners can cause performance degradation
Discussion...

- While in this paper, they assumed that since SVMs were either powerful/useless learners, the normal retraining step in Adaboost wasn’t needed, they have a related follow-up work (Attribute and Simile Classifiers for Face Verification, N. Kumar, A. Berg, P. Belhumeur, S. Nayar. ICCV 2009) where they use forward feature selection instead of Adaboost.

- While they don’t get much better results, their system isn’t restricted to only frontal poses.
Train Classifiers

Pool of Classifiers

Train Classifiers

Pool of Classifiers

Eyes
Mean-Normalized RGB

Train Classifiers

Pool of Classifiers

Train Classifiers

Pool of Classifiers

Select Classifiers

Pool of Classifiers

Selected Classifiers

Error Rate

Iteration

Feature Selection: Smiling

1. Mouth: RGB, Mean Norm., No Aggreg. (M:r.m.n)
2. Mouth: RGB, No Norm., No Aggreg. (M:r.n.n)
3. Mouth: RGB, Energy Norm., No Aggreg. (M:r.e.n)
4. Whole Face: Intensity, No Norm., No Aggreg. (W:i.n.n)
5. ...

Selected Features

Smiling

Selected Features

Gender

Selected Features

Indoor/Outdoor

Selected Features

Hair Color

One global SVM to rule them all...

- The **drawback** of such a local SVM based architecture is that it requires **keeping a large number of SVMs in memory and evaluating all** of them for every new input image...

- This is solved by training **one global SVM** on the **union** of the features from the top N highest weighted (by Adaboost) SVMs.

- The next slide shows the comparative error rates for the local v/s global SVM approach...
Discussion...

• While concatenating the local feature sets (for creating the global SVM from the local top ranked SVMs) they do not seem to use the Adaboost weights/scores for those feature-region sets. Could this help?
## Classification Accuracy

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Error Rates for Attribute-Tuned Local SVMs</th>
<th>Error Rates for Attribute-Tuned Global SVM</th>
<th>Top Feature Combinations in Ranked Order</th>
</tr>
</thead>
</table>
| Age         | 17.34%                                    | 16.65%                                     | W:imn| W:i:n| H:re| E:rn| W:o
| Race        | 7.75%                                     | 6.49%                                      | W:imn| E:re| C:on| M:rn| W:o
| Hair Color  | 7.85%                                     | 5.54%                                      | H:rn| W:imn| E:rn| H:rn| H:rn|
| Mustache    | 6.42%                                     | 4.61%                                      | U:re| M:rn| M:rn| M:re|
| Smiling     | 4.60%                                     | 4.60%                                      | M:rn| M:nn| M:re| M:in| M:in|
| Blurry      | 3.94%                                     | 3.41%                                      | W:mm| H:mm| W:mm| H:mm| M:mm|
| Lighting    | 2.82%                                     | 1.61%                                      | W:i:n| W:ie| K:rn| C:on| E:o
| Environment | 12.25%                                    | 12.15%                                     | N:rn| K:re| K:rn| W:rn| E:re|

Source of slide: The corresponding ECCV 2008 paper
## Comparison to State-of-the-Art

<table>
<thead>
<tr>
<th>Method</th>
<th>Gender Error Rate</th>
<th>Smiling Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>8.62%</td>
<td>4.60%</td>
</tr>
<tr>
<td>Adaboost (pixel comparison feats)</td>
<td>13.13%</td>
<td>7.41%</td>
</tr>
<tr>
<td>Baluja &amp; Rowley, IJCV 2007</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adaboost (Haar-like features)</td>
<td>12.88%</td>
<td>6.40%</td>
</tr>
<tr>
<td>Shakhnarovich et al., ICAFGR 2002</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full face SVM</td>
<td>9.52%</td>
<td>13.54%</td>
</tr>
<tr>
<td>Moghaddam &amp; Yang, TPAMI 2002</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Discussion...

• This approach seems to boil down to feature selection. Could their improved results be because of their rich set of features rather than the Boosting+SVMs approach?
FaceTracer Engine...

- Step1: [Offline] Training attribute-tuned global SVM classifiers for each of the 10 attributes
- Step2: [Offline] All the images in the database are sent through these classifiers to be labeled
- These labels are stored for fast **online** search.
- The search interface accepts simple text based queries and maps these onto the labels by sing a dictionary of terms.
- It returns the results in the order of decreasing confidence.
- Applications: law enforcements, social networking sites, personal snap collection management
Discussion...

• They mention that for multiple-attribute query terms, they convert the classifier confidences into probabilities and then use the product of these probabilities for scoring/ranking. Is this the right approach? Don’t these different attribute classification scores need to be calibrated properly?

• Multi-Attribute Spaces: Calibration for Attribute Fusion and Similarity Search. W. Scheirer, N. Kumar, P. Belhumeur, T. Boult. CVPR 2012
“Asian Babies”

“Adults Outside”

“Middle-Aged White Men”

“Old Men With Mustaches”

“People Wearing Sunglasses Outside”
“Kids Indoors Not Smiling”
“Men With Dark Hair”

“Smiling Asian Men With Glasses”

Personal FaceTracer Search

"Children outside"

Discussion...

• While this method works very well for descriptive attribute based search on facial images, is it scalable to the general type of queries that most image search engines use?
Major contribution of this paper

The idea of allowing people to search for faces with descriptive terms by learning nameable semantic attributes for facial images.
Strengths

• The idea of combining Boosting with SVMs
• This helps on two fronts:
  1. SVMs are powerful classifiers unlike the usual “weak” classifiers that Boosting is used on. This can help where Boosting usually fails.
  2. Boosting helps in selecting the optimal set of features from the variety of feature/region choices available.
• Combining Boosting with SVMs seems to obtain better results than the state-of-the-art approaches that use solely Boosting or SVMs.
Strengths

• Finds an optimal set of relevant features/regions for training for each attribute
• Approach implemented on the largest collection of images of “real-world” faces.
• Easily extensible to new attributes
• Handles both facial attributes & image attributes
Weaknesses

- Limited to frontal poses only
- The methods they compare with do not use the same/similar set of features and instead use a relatively impoverished set of features. As such, they do not seem to be fair baselines
Related work

• "Attribute and Simile Classifiers for Face Verification," (oral presentation)

• Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, Shree K. Nayar, Proceedings of the 12th IEEE International Conference on Computer Vision (ICCV),

• October 2009.
Prior approaches

Images → Low-level features → Verification

RGB → HOG → LBP → SIFT → ...

RGB → HOG → LBP → SIFT → ...

Different

Their approach: attributes

Images → Low-level features → Attributes → Verification

RGB → HOG → LBP → SIFT → ...

RGB → HOG → LBP → SIFT → ...

Male → Asian → Dark hair → Round Jaw

Different

References/Resources


• Attribute and Simile Classifiers for Face Verification, N. Kumar, A. Berg, P. Belhumeur, S. Nayar. ICCV 2009

• Multi-Attribute Spaces: Calibration for Attribute Fusion and Similarity Search. W. Scheirer, N. Kumar, P. Belhumeur, T. Boult. CVPR 2012

• FaceTracer: A Search Engine for Large Collections of Images with Faces. N. Kumar, P. Belhumeur, and S. Nayar. ECCV 2008