Describable Visual Attributes for Face Verification and Image Search

Kumar, Berg, Belhumeur, Nayar. 
*PAMI*, 2011.

Ryan Lei
2011/05/05
Course aMMAI
Outline

• Title explanation
• Introduction
• Related work
• Creating labeled image datasets
• Learning visual attributes
• Application to face verification
• Application to face search
• Possible improvements
Title Explanation

• Describable visual attributes:
  • *Labels* that can be given to an image to describe its appearance.
  • Examples: gender, age, skin color, hair style, smiling.

• Goal 1 - Face verification (authentication):
  a. Determine whether two faces are of the same individual.
  b. Knowing the identity, judge if the query face image is the same person.

• Goal 2 - Image Search:
  • Attribute-based image search system. Sample query: “smiling Asian man with glasses”.
Introduction (1)

• Traditional face recognition research:
  • Use of low-level image features to *directly* train classifier for the classification task.
  • Examples: color (histogram, GCM), texture (Gabor), intensity (Haar, LBP), gradient (SIFT, HOG).
  • The feature representations are often high dimensional, and are in an abstract space.

• Proposal - attribute-based representation:
  a. Describable visual attributes (*“attributes”* in this slide)
  b. *Similes*: similarities to reference faces.
Recap: semantic concept detection (course MMAI).
Concept space for faces -- (a): attributes
Concept space for faces -- (b): similes
Introduction (2)

- Advantages of attribute-based representation:
  - A mid-level representation bridging the “semantic gap”.
  - Dimensionality reduction & manifold discovery.
  - *Flexible*, *generalizable*, *efficient*.

- Flexible: various levels of specificity.
  - “white male” → a set of people.
  - “… + “brown-hair green-eyes scar-on-forehead” → a specific person.
  - “… + “smiling lit-from-above seen-from-left” → a particular image of that person.
Introduction (3)

• Generalizable:
  • Learn a set of attributes from a large image collection, and then apply them in arbitrary combinations to the recognition of unseen images.

• Efficient:
  • $k$ binary attributes can identify up to $2^k$ categories.
  • Requires a much smaller labeled dataset to achieve comparable performance on recognition tasks.
Introduction (4)

- Contributions of this paper:
  1. Introduce attribute & simile classifiers, and face representation using these results.
  2. Application to face verification and face search.
  3. Releases two large public datasets: *FaceTracer* and *PubFig*.

- Previous work:
Creating Labeled Datasets (1)

• Steps 1 & 2:
  • Download from a variety of online resources, such as Yahoo! Images and Flickr.
  • Using a commercial face detector (Omron OKAO), detect faces, pose angles, and fiducial points.
  • Using the yaw angle, flip the face so it always faces left.
  • Align faces using linear least square regression.
Creating Labeled Datasets (1)

- **Steps 1 & 2:**
  - Download from a variety of online resources, such as Yahoo! Images and Flickr.
  - Using a commercial face detector (Omron OKAO), detect faces, pose angles, and fiducial points.
  - Using the yaw angle, flip the face so it always faces left.
  - Align faces using *linear least square regression*. 
Review of linear least square regression

• Basics (1-dimensional-output case):
  • Given a vector $\mathbf{x}$, when want to predict its output $y$.
  • Prepare a training set of $N$ data: $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$.
  • Introduce $M$ basis functions $\Phi_j(.)$, e.g., Gaussian, such that $y$ is a linear combination of $M$ basis components:

$$ y = \sum_{j=1}^{M} w_j \cdot \phi_j(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) $$

• Goal: learn the weight vector $\mathbf{w}$.
• Prediction: $y_{new} = \mathbf{w}^T \phi(\mathbf{x}_{new})$
Review of linear least square regression

• Formulation:
  - N Overdetermined systems: \( \sum_{j=1}^{M} \phi_j(x_i) \cdot w_j = y_i, \quad i = 1, 2, \ldots, N \)

\[
X = \begin{pmatrix}
\phi_1(x_1) & \phi_2(x_1) & \cdots & \phi_M(x_1) \\
\phi_1(x_2) & \phi_2(x_2) & \cdots & \phi_M(x_2) \\
\vdots & \vdots & \ddots & \vdots \\
\phi_1(x_N) & \phi_2(x_N) & \cdots & \phi_M(x_N)
\end{pmatrix}, \quad y = \begin{pmatrix}
y_1 \\
y_2 \\
\vdots \\
y_N
\end{pmatrix} \quad \Rightarrow \quad Xw = y
\]

• Solve the normal equations for \( w \):

\[
X^T X w = X^T y \quad \Rightarrow \quad w = (X^T X)^{-1} X^T y
\]
Review of linear least square regression

- **Extension to D-dimensional-output case:**

\[
W = \begin{pmatrix}
w_{11} & w_{12} & \cdots & w_{1D} \\
w_{21} & w_{22} & \cdots & w_{2D} \\
& \cdots & \ddots & \cdots \\
w_{M1} & w_{M2} & \cdots & w_{MD}
\end{pmatrix},
Y = \begin{pmatrix}
y_{11} & y_{12} & \cdots & y_{1D} \\
y_{21} & y_{22} & \cdots & y_{2D} \\
& \cdots & \ddots & \cdots \\
y_{N1} & y_{N2} & \cdots & y_{ND}
\end{pmatrix}
\]

\[
X^T X W = X^T Y \Rightarrow W = (X^T X)^{-1} X^T Y
\]

- Given 6 fiducial points, predict 6 new coordinates.
- In our case, \( W \) matrix is an *affine transformation*. 
Creating Labeled Datasets (2)

• Step 3: Crowd-sourcing part.
  • They chose Amazon Mechanical Turk, a service that matches workers to online jobs created by requesters.
  • Quality control by requiring confirmation of results by several workers, or minimum worker experience, etc.
  • After verification, they collected 145,000 attribute labels, at the cost of USD$6,000.
Amazon Mechanical Turk

Worker

Requester
Creating Labeled Datasets (3)

• FaceTracer – attribute labels:
  • 15,000 faces with 5,000 labels. (0.33 labels per image)
  • Researchers still need to train their own attribute classifiers to transform the faces into attribute space.
  • http://www.cs.columbia.edu/CAVE/databases/facetracer/

• PubFig – identity labels:
  • 58,797 images of 200 public figures.
  • Development set: 60 people. (for training simile classifiers)
  • Evaluation set: 140 people. (same purpose as LFW)
  • http://www.cs.columbia.edu/CAVE/databases/pubfig/

• What if jointly labeling identities and attributes?
# of faces of this person.

(a) PubFig Development set (60 individuals)

(c) All 170 images of Steve Martin
Attributes labeled in the FaceTracer dataset

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>gender</td>
<td>male, female</td>
</tr>
<tr>
<td>race</td>
<td>asian, white, black</td>
</tr>
<tr>
<td>age</td>
<td>baby, child, youth, middle_aged, senior</td>
</tr>
<tr>
<td>hair_color</td>
<td>blond, not_blond</td>
</tr>
<tr>
<td>eye_wear</td>
<td>none, eyeglasses, sunglasses</td>
</tr>
<tr>
<td>mustache</td>
<td>true, false</td>
</tr>
<tr>
<td>expression</td>
<td>smiling, not_smiling</td>
</tr>
<tr>
<td>blurry</td>
<td>true, false</td>
</tr>
<tr>
<td>lighting</td>
<td>harsh, flash</td>
</tr>
<tr>
<td>environment</td>
<td>outdoor, indoor</td>
</tr>
</tbody>
</table>

However, all attribute classifiers in this work are binary.
Learning Visual Attributes (1)

• Definitions:
  • Consider the attribute “gender” and the qualitative labels “male” and “female”.
  • An Attribute can be thought of as a function that maps an image $I$ to a real value $r_i$.

- Female - 0 - Male

• Large positive (negative) values of $r_i$ indicate the present (absent) of the $i$-th attribute.

• Core part in this work:
  • Feature selection, though still an open problem in machine learning.
Learning Visual Attributes (2)

- Low-level features options:
  a) Region of face. [10]
  b) Type of pixel value. [5]
  c) Normalization to apply. [3]
  d) Level of aggregation to use. [3]
  - A total of 10*5*3*3 = 450 combinations.
Learning Visual Attributes (3)

- Complete feature pool:
  - However, not all possible combinations are valid, e.g., normalization of hues.

\[ \hat{x} = \frac{x}{\mu} \]

\[ \hat{x} = \frac{x - \mu}{\sigma} \]

<table>
<thead>
<tr>
<th>Pixel Value Types</th>
<th>c) Normalizations</th>
<th>d) Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB</td>
<td>None</td>
<td>None (concat)</td>
</tr>
<tr>
<td>HSV</td>
<td>Mean Normalization</td>
<td></td>
</tr>
<tr>
<td>Image Intensity</td>
<td>Energy Normalization</td>
<td></td>
</tr>
<tr>
<td>Edge Magnitude</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edge Orientation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
• **Forward feature selection** – in each iteration:
  1. Train all possible classifiers by *concatenating* one feature option with the current feature set.
  2. Evaluate the performances by cross-validation.
  3. Feature(s) with highest CV accuracy → added.
  4. Drop the lowest-scoring 70% of features.
  5. Keep adding until the accuracy stops improving.
Learning Visual Attributes (5)

- Classifiers used:
  - SVMs with RBF kernel. (tool: libSVM)
  - Seems to be only one final classifier.
  - ECCV 2008: Boosting approach.

- Thought:
  - Features A and B work well individually. Will they work well together?
Typically range from 80% to 90%.

Analysis shows that all regions and feature types are useful: the power of feature selection.
Learning Visual Attributes (6)

- Simile classifiers:

<table>
<thead>
<tr>
<th>Simile</th>
<th>Positive Examples</th>
<th>Negative Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reference Person 1</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eyebrows</td>
<td><img src="image1" alt="Positive Examples" /></td>
<td><img src="image2" alt="Negative Examples" /></td>
</tr>
<tr>
<td>Eyes</td>
<td><img src="image3" alt="Positive Examples" /></td>
<td><img src="image4" alt="Negative Examples" /></td>
</tr>
<tr>
<td>Nose</td>
<td><img src="image5" alt="Positive Examples" /></td>
<td><img src="image6" alt="Negative Examples" /></td>
</tr>
<tr>
<td>Mouth</td>
<td><img src="image7" alt="Positive Examples" /></td>
<td><img src="image8" alt="Negative Examples" /></td>
</tr>
<tr>
<td><strong>Reference Person 2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eyebrows</td>
<td><img src="image9" alt="Positive Examples" /></td>
<td><img src="image10" alt="Negative Examples" /></td>
</tr>
<tr>
<td>Eyes</td>
<td><img src="image11" alt="Positive Examples" /></td>
<td><img src="image12" alt="Negative Examples" /></td>
</tr>
<tr>
<td>Nose</td>
<td><img src="image13" alt="Positive Examples" /></td>
<td><img src="image14" alt="Negative Examples" /></td>
</tr>
<tr>
<td>Mouth</td>
<td><img src="image15" alt="Positive Examples" /></td>
<td><img src="image16" alt="Negative Examples" /></td>
</tr>
</tbody>
</table>
Learning Visual Attributes (6)

- Simile classifiers:
  - 60 reference people.
  - 8 manually selected face regions. (Similes are component-based by definition.)
  - 6 manually selected combinations of [pixel value type, normalization, aggregation], *without automatic selection*. (Could have done so.)
  - Yield a total of $60 \times 8 \times 6 = 2,880$ simile classifiers, each being an SVM with RBF kernel.
• Now we have these attributes and similes as mid-level representation.

• How do we apply them in high-level classification / retrieval problems?
Application to Face Verification (1)

• Problem:
  • “Are these two faces of the same person?”

• Existing methods:
  • Very early work & early work:
    • Compare L2 distance in PCA-reduced space.
    • Improved by the supervised LDA.
    • Early work used well-known low-level features directly.
  • Often make *avoidable* mistakes: men being confused for women, young people for old, Asians for Caucasians.
  • From this observation, they claim that the attribute and simile classifiers can avoid such mistakes.
Application to Face Verification (2)

- Many steps have been explained in previous sections.
- Goal here: the verification classifier.
Application to Face Verification (3)

- Verification classifier:
  - Train a verification classifier \( V \) that compares attribute vectors \( C(I_1) \) and \( C(I_2) \) of two face images \( I_1 \) and \( I_2 \), and returns the decision \( v(I_1, I_2) \).
  - Define \( a_i = C_i(I_1) \) and \( b_i = C_i(I_2) \), subscript \( i \) meaning the \( i \)-th attribute (simile) value. \( i = 1, 2, \ldots, n \).
  - Use both the absolute difference and product.
    1) \( |a_i - b_i| \): Observation that \( a_i \) and \( b_i \) should be close if they are of the same individual.
    2) \( a_i * b_i \): Observation that \( a_i \) and \( b_i \) should have the same sign if they are of the same individual.
Application to Face Verification (4)

- Verification classifier (cont.):
  - Concatenate them in tuple \( p_i \), then concatenate for all \( i \):
    \[
    p_i = \langle |a_i - b_i|, a_i b_i \rangle \quad v(I_1, I_2) = V(\langle p_1, \ldots, p_n \rangle)
    \]
  - \( \langle p_1, \ldots, p_n \rangle \) is the training input to the classifier.
  - Again, they use SVM with RBF kernel.

- Experiment:
  - Performed on datasets LFW and PubFig.
  - Even the *individuals* are disjoint in training / testing sets.
  - In other words, machine never sees the same pair of people in its model. Rely solely on attributes / similes.
Fig. 10. Face verification performance on LFW
How well can attributes potentially do?

- Attributes obtained by machine learning: 81.57%.
- Attributes obtained by human labeling: 91.86%.
- Face verification entirely by human: 99.20%.
Application to Face Search

• FaceTracer search engine:
  • For details, refer to the work in ECCV 2008.
  • Demo video on YouTube: http://www.youtube.com/watch?v=20UJ7JL7RNs
  • Text-based query. Remember that we have bridged the semantic gap by those mid-level classifiers.
  • For each attribute, search results are ranked by confidences.
  • Convert confidences into probabilities in $[0,1]$ by Gaussian.
  • For multiple query terms, just combine them by taking the product (AND) of those probabilities.
• Indexing: basic inverted index? Not mentioned.
What about the *user gap*?
To accurately capture user’s search intention:

- **Sketch-based**
  - Example: A couple, Sunset, Mountain, Sea

- **Concept-based**
  - Example: Jeep, Grass

- **People-attribute-based**
  - Example: Icons of people with attributes
Possible Improvements

• Go beyond binary classifiers for attributes.
  • Regressors are good for quantitative attributes like “age”.

• Use feature selection in training the Simile classifiers.

• Automatic (Dynamic) selection of attributes.

• Image search:
  • Product of probabilities can’t prevent outlier scores.
  • TF-IDF approach for ranking the “word frequencies”?
  • Combine identity information?
  • Combine location and size, even sketch?
Thank You!

Q & A