Lookalike Disambiguation in Face Recognition

Arun Ross
Professor
Michigan State University
rossarun@cse.msu.edu
http://iprobe.cse.msu.edu

Work done with Thomas Swearingen
In this presentation we are **not** considering “lookalike faces” from a human vision standpoint

**Not** specifically considering twins, siblings, and other types of kinship relationships

We are considering “lookalike faces” from a **computer vision** standpoint

- Face images of different identities that are “confused” to be the same by a **face matcher**
Examples of Lookalike Faces

- LFW Dataset | COTS Matcher

- Images with similarity scores:
  - Score: 0.99
  - Score: 0.85
  - Score: 0.92
  - Score: 0.82
Most face recognition methods do not explicitly consider the notion of similarity during the training phase.

Face images are labeled with identifiers:

001; 001; 001; 002; 002; 003; 003; 003; 003; 003

Then the method attempts to minimize intra-class variations and maximize inter-class variations.

During this process, the degree of similarity between different identities is not explicitly used – since that information is typically not available during training.
Training Stage

Training Set

- The distance between different identities is not explicitly specified during the training phase; it is implicitly learned by the face matcher.
- But see: Sadovnik, *Finding your Lookalike: Measuring Face Similarity Rather than Face Identity*, CVPRW 2018
Identification Process

\[ d = 0.9 \quad d = 1.3 \quad d = 1.4 \quad d = 0.6 \quad d = 0.8 \quad d = 0.85 \quad d = 0.58 \]
Ranked Match List

GALLERY:

PROBE:

Ranked Match List

r=0.58  r=0.6  r=0.8  r=0.85  r=0.9  r=1.3  r=1.4

Rank 1  Rank 2  Rank 3  Rank 4  Rank 5  Rank 6  Rank 7
Correct match occurs at rank 2, not rank 1

Matcher “confuses” imposter face at rank 1 with genuine face at rank 2

These 2 face could be lookalikes
## Related Work

<table>
<thead>
<tr>
<th>Year</th>
<th>Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td>Srinivas: Analysis of facial marks to distinguish between identical twins</td>
</tr>
<tr>
<td>2012</td>
<td>Le: A facial aging approach to identification of identical twins</td>
</tr>
<tr>
<td>2018</td>
<td>Sun: Deep Siamese convolutional neural networks for identical twins and look-alike identification</td>
</tr>
<tr>
<td>2017</td>
<td>Smirnov: Doppelgänger mining for face representation learning</td>
</tr>
<tr>
<td>2017</td>
<td>Moeini: Open-set face recognition across look-alike faces in real-world scenarios</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use facial marks to distinguish twins</td>
</tr>
<tr>
<td>Face aging to distinguish twins</td>
</tr>
<tr>
<td>Develop CNN to distinguish twins</td>
</tr>
<tr>
<td>Match face regions independently</td>
</tr>
<tr>
<td>Refine mini-batch selection of a general-purpose matcher using a list of lookalikes</td>
</tr>
<tr>
<td>Use 3D models to distinguish lookalikes</td>
</tr>
</tbody>
</table>

- Identical twins were an early interest where approaches focused on a specific aspect of the face.
- Later approaches, focused on lookalikes more generally.
Choose some of the top ranked faces on initial ranked match list to re-rank

- Re-rank them using a lookalike disambiguator (LD)
  - LD matcher specifically trained to distinguish lookalike face images
Selecting Gallery Samples to Re-rank

- Conduct an analysis to determine how the scores in the ranked match list vary in the vicinity of a correct match.
Match-Vicinity Analysis

• Find the **match-vicinity scores** for a given probe image $p$ in a ranked match list

• Normalize score with respect to the score at **position of correct gallery match** $(d_p^{(c)})$

  $$s_p^{(i)} = d_p^{(i)} - d_p^{(c)}$$

• Normalized score

  • **before** rank $c$ must be **non-positive**
  • **after** rank $c$ must be **non-negative**
Match-Vicinity Analysis

Correct Match
Rank c

\[ d_p^{(c-1)} \]
\[ d_p^{(c)} \]
\[ d_p^{(c+1)} \]

\[ s_p^{(c-1)} = d_p^{(c-1)} - d_p^{(c)} \]
\[ s_p^{(c+1)} = d_p^{(c+1)} - d_p^{(c)} \]

\[ s_p^{(c)} = d_p^{(c)} - d_p^{(c)} = 0 \]
Match-Vicinity Plot (MVP)

- MVP shows **mean** and **SD** of normalized match scores in the match vicinity for 3,728 probe images queries
- **Dataset:** TinyFace dataset | **Matcher:** ArcFace matcher
Adaptive Re-Ranking

- MVP: distance score increases at a **higher rate** from one rank to the next **after** encountering the correct match
- Use **sharp increases** in distance score to determine subset selection
Adaptive Re-Ranking

- Given a probe image, $p$, and a gallery set, $\mathcal{G} = \{g_1, g_2, ..., g_n\}$

- Compare $p$ to each gallery image $g_i$ to obtain ranked list, $\mathcal{L} = (d^{(1)}, d^{(2)}, ..., d^{(n)})$

- Calculate **rolling sum** over consecutive distance scores, $S_k$

- Re-rank the **top $k$** matches
  - the smallest value of $k$ such that $S_k > \tau$
General Purpose Matcher (GPM)

• **ArcFace** is a publicly-available face matcher
  • High performance on LFW dataset (99.8% accuracy)

• Outputs a **512-dimensional** representation for a given input image

• Compare representations using **Euclidean distance**

Deng et al., “Arcface: Additive angular margin loss for deep face recognition,” CVPR 2019
Lookalike disambiguator (LD)

• **Finetunes** GPM using lookalike **triplets**

• Lookalike triplet consists of **anchor**, **positive**, and **negative** samples
  
  • **Anchor & positive** sample of same subject
  
  • **Anchor & negative** samples of different subjects, but judged by GPM to be **lookalikes**

• **Loss function**

\[
L = \sum_{\{I_a, I_p, I_n\}} \|f(I_a) - f(I_p)\|_2 - \|f(I_a) - f(I_n)\|_2 + \alpha \text{margin}
\]
2 triplets from 1 pair

(Anchor, Positive, Negative)

(Anchor, Positive)

(Anchor, Positive, Negative)

Lookalike Triplet 1

Lookalike Pair

Lookalike Triplet 2
Lookalike disambiguator (LD)

Training Parameters

- PyTorch environment
- Stochastic gradient descent with Adam optimizer
- $\alpha_{\text{margin}} = 0.2$
- Batch Size: 32
- Learning Rate: 0.01
TinyFace Dataset

• Dataset consisting of **small** face images
  • Average size 20 x 16 pixels
• **Gallery-match** and **Probe** sets used
• Gallery contains multiple images of the same subject
• Identification experiments are **closed-set**

<table>
<thead>
<tr>
<th>Set</th>
<th>Num. Images</th>
<th>Num. Subjects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probe</td>
<td>3,728</td>
<td>2,569</td>
</tr>
<tr>
<td>Gallery-Match</td>
<td>4,443</td>
<td>Unknown</td>
</tr>
<tr>
<td>Gallery-Distractor</td>
<td>153,428</td>
<td>Unknown</td>
</tr>
</tbody>
</table>
TinyFace Dataset

Cheng et al., “Low-resolution face recognition,” ACCV 2018
Filtered TinyFace Dataset

• Dataset manually filtered to exclude profile-view faces

• Filtered dataset contains 1,145 subjects
  • 2,081 images in probe subset
  • 2,461 images in gallery subset

• Experiments conducted on filtered dataset
Lookalike Discovery

- Match **gallery** against itself using **GPM**
- Select **imposter pairs** in the distance score range $[0, 0.8]$
- Results in $\sim 679K$ lookalike pairs
  - $6.9\%$ of all imposter pairs

\[
p(\text{score}|\text{genuine}) \gg p(\text{score}|\text{imposter})
\]
Evaluation Metrics

Re-rank Subset Selection

1. Hit Rate
   Fraction of probes for which the selection scheme chooses a gallery subset that includes the correct match

2. Surplus Size
   Number of samples included in the subset with rank higher than the rank of the correct match

3. Pool Size
   Number of gallery samples selected to be reranked
Parameter Selection (using gallery)

- Estimate $q$ and $\tau$ from gallery dataset (filtered)
- Rolling sum calculated for those gallery samples that have at least 1 other gallery sample of the same subject
  - 1,897 such images
- $\tau$ is the average value of the rolling sum taken at position of correct match ($S_c$)

<table>
<thead>
<tr>
<th>$q$</th>
<th>$\tau$</th>
<th>Surplus Size</th>
<th>Hit Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Total</td>
<td>Per Search</td>
</tr>
<tr>
<td>1</td>
<td>0.7695</td>
<td>270,276</td>
<td>142.5</td>
</tr>
<tr>
<td>2</td>
<td>1.378</td>
<td>294,003</td>
<td>155.0</td>
</tr>
<tr>
<td>3</td>
<td>1.958</td>
<td>295,173</td>
<td>155.6</td>
</tr>
<tr>
<td>4</td>
<td>2.511</td>
<td>296,353</td>
<td>156.2</td>
</tr>
<tr>
<td>5</td>
<td>3.049</td>
<td>297,541</td>
<td>156.8</td>
</tr>
<tr>
<td>6</td>
<td>3.574</td>
<td>298,737</td>
<td>157.5</td>
</tr>
<tr>
<td>7</td>
<td>4.090</td>
<td>299,942</td>
<td>158.1</td>
</tr>
<tr>
<td>8</td>
<td>4.597</td>
<td>301,152</td>
<td>158.8</td>
</tr>
<tr>
<td>9</td>
<td>5.094</td>
<td>302,365</td>
<td>159.4</td>
</tr>
<tr>
<td>10</td>
<td>5.584</td>
<td>303,583</td>
<td>160.0</td>
</tr>
</tbody>
</table>
Fixed versus Adaptive

• Compare Fixed and Adaptive selection schemes

• For adaptive scheme, $q = 10$ and $\tau = 5.584$

• For fixed scheme, top 10% of matches are reranked (246)

• A small pool size is generally better
  • Not inherently bad: Could be that correct match occurs at a higher rank
Fixed versus Adaptive

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Pool Size (min/mean/median/max)</th>
<th>Hit Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed</td>
<td>246</td>
<td>246</td>
</tr>
<tr>
<td>Adaptive</td>
<td>5</td>
<td>20.66</td>
</tr>
</tbody>
</table>
Identification Performance

- Given a probe: Use GPM to rank gallery samples
- Select gallery samples to re-rank using fixed and adaptive schemes
- Re-rank top gallery samples using LD

Rank-1 identification accuracy improves from ~40.7% to ~49.6%
Summary

- Proposed an adaptive gallery selection scheme based on match scores generated using a face matcher.

- Proposed the use of a separate matcher for re-ranking lookalike face images.

- Observed an improvement in identification accuracy when using a Lookalike Disambiguator on the selected gallery samples.

Preliminary results presented; Experiments with other datasets and matchers are ongoing; Motion can help in disambiguating as well.
Lookalike Disambiguation in Face Recognition

Arun Ross
Professor
Michigan State University
rossarun@cse.msu.edu
http://iprobe.cse.msu.edu

Work done with Thomas Swearingen