Automatic face naming using caption-based supervision

M. Guillaumin, T. Mensink, J. Verbeek, C. Schmid
LEAR, INRIA, Grenoble, France; contact: matthieu.guillaumin@inria.fr, web: http://lear.inrialpes.fr/

Summary
- Methods for automatic face naming:
  - to find faces of a given person;
  - to associate the detected faces in the images with the extracted names in the captions.
- Supervision consists of news images with descriptive captions, see examples.
- Supervision is weak and automatic: names not appearing in the captions are prevented from being assigned to faces found in the corresponding images.
- Uses a graph of similarities between faces.
- State-of-the-art performance on the Yahoo! News dataset for both tasks.

Task 1 – Single-person query
To retrieve the faces of a given person in a news dataset. The query is text-based, e.g. “David Beckham”.

Method
- Initial text-based query: keep documents only if their captions match the query.
- Detect faces in retrieved images using Mikolajczyk [1].
- Compute the similarity graph for the faces (see left).
- Find the densest component in the resulting similarity graph (see below).

Densest component search with constraints
The densest component is a subgraph $S$ with high edge weights $w_{i,j}$ formally defined as maximizing:

$$f(S) = \sum_{i \neq j} w_{i,j}$$

Baseline: a greedy search, $S$ starts as the whole graph, and at each iteration the least connected face is removed and $f(S)$ is re-evaluated. The subset with highest encountered density is kept. Ozkan et al [2].

Document constraint: the greedy search is adapted by starting with only the most connected face within each image (DC).

Local search: added after DC, iteration over documents is performed to select the best face in each document.

Experiments and results
Performance is measured on 23 queries over 15000 stories. Shown are the precisions at 85% recall for individual queries using the baseline method (IP-AV-GR), and our best method (IP-CT-KNN-DC+LS). The queries were sorted by the precision of the text-based result.

Task 2 – Multi-person naming
To name all the faces in images of a news dataset.

Method
- Extract named entities: captions are processed using a CRF-based Named Entity Recognizer from Deschacht [3] to obtain candidate names for the detected faces.
- Detect faces in all the documents using Mikolajczyk [1].
- Compute the similarity graph for all faces (see left).
- Find non-overlapping clusters that have highest inner similarities by iteratively considering all documents (see below).

Constrained similarity clustering
Clusters are subgraphs $(S_n)$ of the similarity graph, one for each name $n$. We try to find the set of clusters maximizing the inner-similarity and complying to the constraints given by documents:

$$F(\{S_n\}) = \sum_{n} \sum_{i \neq j} w_{i,j}$$

This is intractable, so an approximate method is proposed: the subgraphs are optimized for each document iteratively, until convergence. Shown is an example of a document with faces $f_1$ and $f_2$ and three names corresponding to subgraphs $S_1$, $S_2$ and $S_3$. Given the sum of edge weights (represented by width) that connect each face to the clusters, we search for the best admissible assignment.

Baseline: Generative Mixture Model, Berg et al [4]. Refer to as (Gen) or (GenPP) when pre-processing is done (PCA, LDA). Model allocates one gaussian for each name in the feature space.

Experiments and results
We use a fully annotated dataset of 857 documents (1183 detected faces, 1528 named entities, 424 unique names). We evaluate performance using two measures:
- The accuracy is the percentage of correct assignments for all the faces, including assignments to null.
- Precision is the percentage of correct assignments among the faces that were assigned to a name. Some of our naming algorithms are compared below, with markers where the best accuracy is obtained.

References

Comparison by averaging the precisions over 23 queries.

<table>
<thead>
<tr>
<th>Recall / Precision (%)</th>
<th>Baseline r-neighborhood</th>
<th>IP-AV-GR</th>
<th>IP-CT-KNN-DC+LS</th>
<th>kNNGPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td>62.8</td>
<td>73.4</td>
<td>68.4</td>
<td>70.8</td>
</tr>
<tr>
<td>85</td>
<td>66.1</td>
<td>73.6</td>
<td>68.2</td>
<td>72.4</td>
</tr>
</tbody>
</table>

Average Processing Time

<table>
<thead>
<tr>
<th>Number of faces × Number of names</th>
<th>Brute-force Algorithm</th>
<th>Max-Flow Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>1000</td>
<td>500</td>
</tr>
</tbody>
</table>

Similarity graph of faces
Similarities between faces are computed over pairings of SIFT features, either at Difference-of-Gaussians interest points (IP) or facial features (FF).

A pairing is obtained when SIFT features are closest in the feature space, and points geometrically consistent. The considered similarity measures are:
- average distance between paired features (AV);
- number of paired features (CT).

The resulting graph is optionally transformed by:
- thresholding the similarities (r-neighborhood);
- keeping only the $k$-nearest neighbors (kNN);
- linearly soft-thresholding the similarities (LT).

Local search: added after DC, iteration over documents is performed to select the best face in each document.