Automatic face naming using caption-based supervision

M. Guillaumin, T. Mensink, J. Verbeek, C. Schmid

LEAR team, INRIA, Grenoble, France, contact: `matthieu.guillaumin@inria.fr`, web: `http://lear.inrialpes.fr/`

Summary

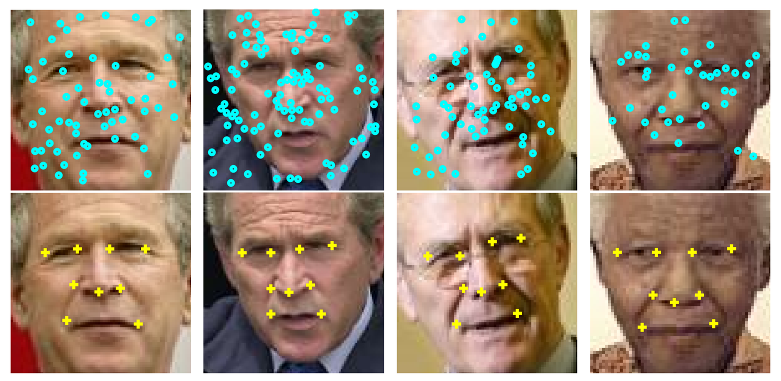
- Methods for automatic face naming:
  - to find faces of a given person;
  - to associate the detected faces in the images with the extracted names in the captions.
- Dataset consists of news images with descriptive captions, see examples.
- Supervision is weak and automatic: names not appearing in 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.

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).

IP

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 (ϵ-neighborhood);
- keeping only the k-nearest neighbors (kNN);
- linearly soft-thresholding the similarities (LT).

Comparison by averaging the precisions over 23 queries.

Recall	Baseline	IP-AV	IP-CT	FF-AV	FF-CT	IP-AV	IP-CT	FF-AV	FF-CT
		ϵ-neighborhood				kNN linear			
75	68.2	71.3	73.8	67.2	69.6	64.7	77.6	73.7	74.1
85	62.8	66.1	68.4	62.6	63.1	63.5	73.0	70.8	71.5

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 height edge weights w_{ij}, formally defined as maximizing :

$$f(S) = \frac{\sum_{i,j \in S} w_{ij}}{|S|}. \quad (1)$$

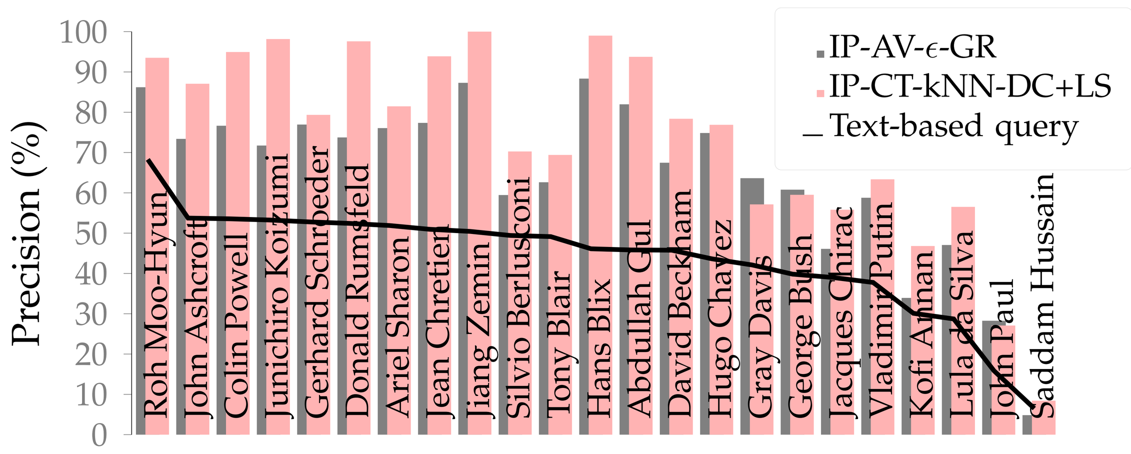
Baseline: a greedy search. S starts as the whole graph, and at each iteration the least connected face is removed and $f(S)$ is re-evaluated. The subset with highest encountered density is kept. Ozkan *et al* [2].

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-ϵ-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.

German Chancellor **Angela Merkel** shakes hands with Chinese President **Hu Jintao** (…)

Kate Hudson and **Naomi Watts**, Le Divorce, Venice Film Festival - 8/31/2003.

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).

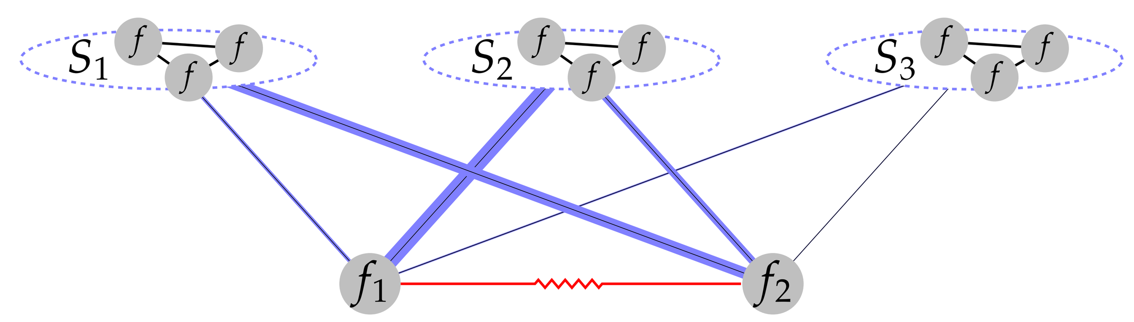
Constrained similarity clustering

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:

$$F(\{S_n\}) = \sum_n \sum_{i,j \in S_n} w_{ij}. \quad (2)$$

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]. Refered to as (Gen) or (GenPP) when pre-processing is done (PCA, LDA). Model allocates one gaussian for each name in the feature space.



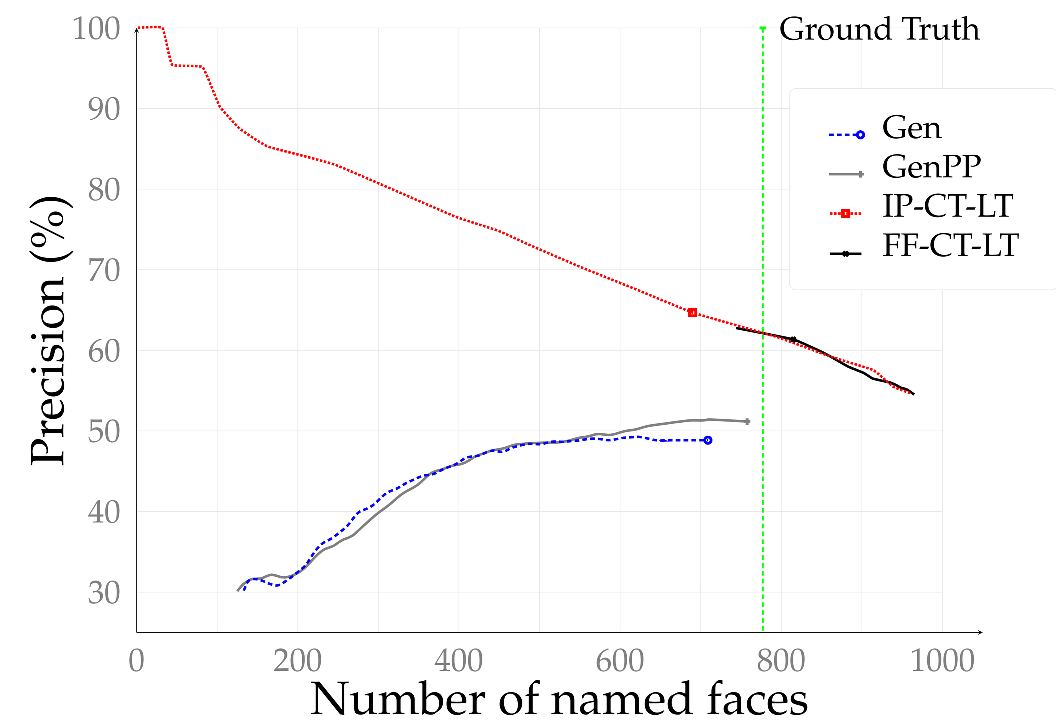
Experiments and results

We use a fully annotated dataset of 857 documents (1183 detected faces, 1528 named entities, 424 unique names). We evaluate performance using two measures:

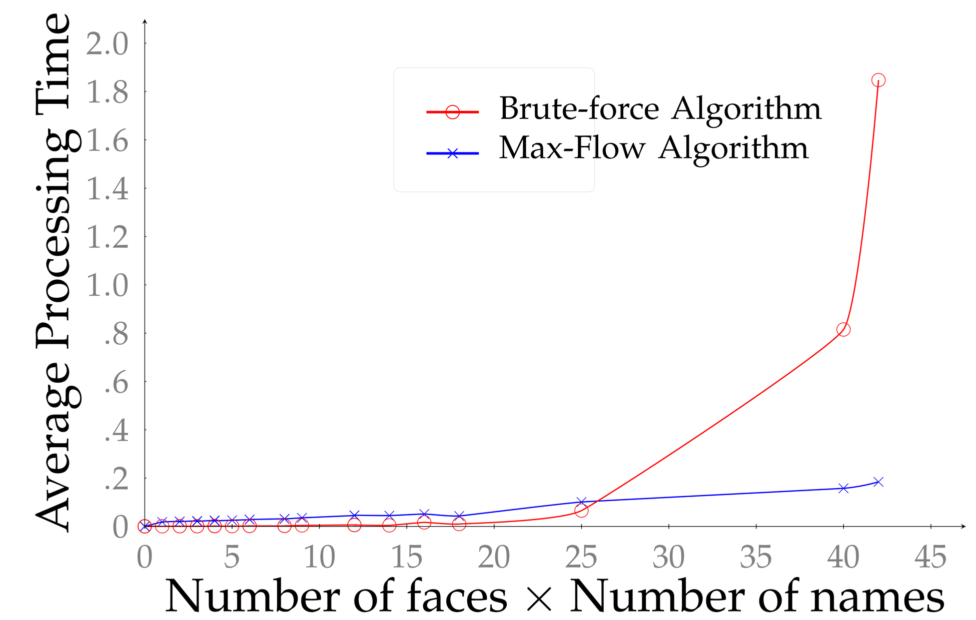
- The accuracy is the percentage of correct assignments for all the faces, including assignments to *null*.

Gen	GenPP	DENS	HT	kNN	HT	LT	LT
	FF	FF-CT	FF-AV	FF-CT			IP-CT
48.7	51.7	40.1	59.3	59.1	63.4	63.5	62.0

- 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.



The number of assignments to consider grows exponentially with the number of faces and names. The proposed Max-flow algorithm keeps computation times reasonable.



References

[1] K. Mikolajczyk and C. Schmid and A. Zisserman Human detection based on a probabilistic assembly of robust part detectors ECCV, 69–81, 2004.

[2] D. Ozkan and P. Duygulu. A Graph Based Approach for Naming Faces in News Photos. CVPR, 1477–1482, 2006.

[3] K. Deschacht and M. Moens. Efficient Hierarchical Entity Classification Using Conditional Random Fields. Workshop on Ontology Learning and Population, 2006.

[4] T. Berg *et al.* Names and Faces in the News. CVPR, 848–854, 2004.