Automatic Face Naming with Caption-based Supervision

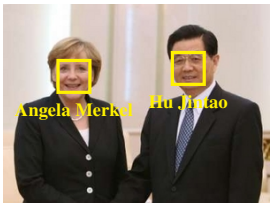
Matthieu Guillaumin, Thomas Mensink, Jakob Verbeek and Cordelia Schmid
LEAR team, INRIA, Grenoble, France
{matthieu.guillaumin,thomas.mensink,jakob.verbeek,cordelia.schmid}@inria.fr

Abstract

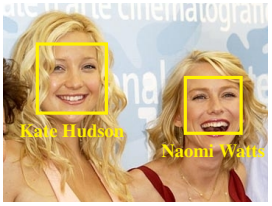
We consider two scenarios of naming people in databases of news photos with captions: (i) finding faces of a single person, and (ii) assigning names to all faces. We combine an initial text-based step, that restricts the name assigned to a face to the set of names appearing in the caption, with a second step that analyzes visual features of faces. By searching for groups of highly similar faces that can be associated with a name, the results of purely text-based search can be greatly ameliorated. We improve a recent graph-based approach, in which nodes correspond to faces and edges connect highly similar faces. We introduce constraints when optimizing the objective function, and propose improvements in the low-level methods used to construct the graphs. Furthermore, we generalize the graph-based approach to face naming in the full data set. In this multi-person naming case the optimization quickly becomes computationally demanding, and we present an important speed-up using graph-flows to compute the optimal name assignments in documents. Generative models have previously been proposed to solve the multi-person naming task. We compare the generative and graph-based methods in both scenarios, and find significantly better performance using the graph-based methods in both cases.

1. Introduction

Over the last decades large digital multimedia archives have appeared, through digitalization efforts by broadcasting services, through news oriented media publishing online, and through user provided content concentrated on websites such as YouTube and Flickr. The work in this paper fits within a broad ongoing effort [2, 10] to develop methods to allow access to such archives in a user-oriented and semantically-meaningful way. The volume of data in such archives is generally large, and the semantic concepts of interest differ greatly between different archives. As a result, there is a great interest in 'unsupervised' systems for automatic content analysis in such archives. These contrast with 'supervised' systems which require manual annotations to link content to semantic concepts.



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



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

Figure 1. Examples of typical image-caption pairs in the Yahoo! News data set, and the result of automatic face naming.

The crux of unsupervised systems is to exploit the relations between different media, such as the relation between images and text, and between video and subtitles combined with scripts [1, 9, 15, 18]. The correlations that can be automatically detected are typically less accurate – *e.g.* images and text associated using a web search engine like Google [4, 10] – than supervised information provided by manual efforts. However, the important difference is that the former can be obtained at a lower cost, and therefore from much larger amounts of data, which may in practice outweigh the higher quality of supervised information.

In this paper we consider two problems: finding among all detected faces those depicting a certain person, and attaching names to all faces appearing in an image. For both tasks we use the Yahoo! News data set, also used in [3]: a data set of roughly 15000 pictures and captions, see Figure 1 for two examples. It contains 15280 detected named entities and 22750 detected faces appearing with wide variations in pose, expression, and illumination from about 1250 different individuals. To find people in such a database, text alone is clearly insufficient: if we return all faces in pictures that have the queried person in the caption we find a precision of 44% (averaging over 23 queries and comparing to a hand-labeled ground truth). Our experimental results show that by including visual information these results can be dramatically improved, consistently with what has been observed for face naming in videos. Everingham *et al*. [9]

rely on multiple cues: captions provide names, audio tracks indicate which character is speaking, tracking ensures time and space consistency. Our setting is more challenging in the sense that we only have still images and captions, which carry less information.

We extend a graph-based method for finding faces of a single person by Ozkan and Duygulu [14]. First, images containing the name of the queried persion are retrieved. As a second step, a similarity graph over all faces detected in these images is constructed. Then an approximate search for the densest component of the graph is performed to select the faces belonging to the queried person. We introduce the constraint that each image may contain the queried person only once, and improve the low-level methods used to construct the graph. These contributions lead to significantly better results: precision is 10% higher for a recall of 85%. For the second problem, assigning names to all faces in the database, a generative mixture model was proposed by Berg *et al*. [3]. The main idea of this approach is to perform a constrained clustering, where clusters are associated with names in the document, and each name may be assigned to at most one face. We extend the graph-based approach to this setting, and compare it to the generative approach. In our experiments we obtain 12% higher accuracy using the graph-based approach. The generative model also performs worse in the single-person case.

For the detection of named entities and faces we use off-the-shelf techniques [8, 13], see examples in Figure 1. Faces are represented using SIFT [12] descriptors on points detected using different strategies, including the one in [9].

The rest of this paper is organized as follows: In Section 2 we propose a series of improvements on the graph-based single-person approach by [14], and in Section 3 we show how the graph-based method extends to the multi-person problem and how we can efficiently optimize the assignments in this case. In Section 4 we present our experimental results, and show the performance increases obtained by our improvements. Section 5 concludes the paper.

2. Single-person Retrieval

The task considered in this section is retrieving all faces of one person. We first limit the search to those documents that contain the name in the caption. Then, we detect faces in these images to create a set of candidate faces. Visual features from these faces can be used to refine the initial result set based on the following assumptions: (i) the number of faces corresponding to the query will be relatively large, (ii) pairs of faces belonging to the queried person are more similar than arbitrary pairs of faces, and (iii) the queried person appears at most once in each image.

As a baseline method we use the approach of [14], which constructs a graph over the faces based on similarities to further analyze the initial result set. In the rest of this section we present the baseline approach and propose a series of modifications. In Section 4 we show experimentally that these lead to significant performance gains. In Section 2.1 we improve the graph analysis by taking account of assumption (iii) and adding a local search technique. Two choices for constructing a graph from pairwise similarities are considered in Section 2.2. Different feature extraction methods and similarity measures are discussed in Section 2.3.

2.1. Document Constrained Densest Component

We define a graph $G = (V, E)$ where the vertices in V represent faces and edges in E are weighted according to similarity between faces. To filter our initial results, we search for the densest subgraph, or *component*, of G. The density $f(S)$ of a component $S \subseteq V$ is defined as the sum of weights w_{ij} for $i, j \in S$, divided by the cardinality of S:

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

The densest component of a graph corresponds to a group of vertices with high edge weights between them, while excluding ones that connect to them with lower edge weights.

The baseline method uses a greedy 2-approximate algorithm which ensures the density of the returned component is in the worst case half of the maximal density [6]. The greedy search starts with the entire graph as subset ($S = V$). At each iteration, $f(S)$ is computed and the node with the minimum sum of edge weights within S is removed. Finally, the subset with the highest encountered density is returned as the densest component.

Note that this greedy search leaves room for improvement, and that the returned subset may contain multiple faces from a single image, contravening assumption (iii). We therefore propose to modify the search to consider only subsets S with at most one face from each image. We initialize this procedure by selecting from each image the face that has the highest sum of edge weights, and then run the greedy algorithm to select a subset of these faces. However, previously rejected faces might now yield higher densities for S than the initial choice. Consequently, we refine the greedy search by performing an additional local search. This local search proceeds by iterating over the images and checking which face in the image yields the highest density, or whether selecting none of the faces leads to a yet higher density. The process terminates when all nodes have been considered without obtaining further increases.

2.2. Graph Construction

The baseline method for constructing a graph over faces from some similarity or distance measure is to apply a threshold ϵ on the distances and to include an edge in the graph whenever the distance is smaller than ϵ. In effect, this

connects every face to all other faces in an ϵ-neighbourhood, and results in a symmetric connectivity. This implicitly assumes that the distance measure is 'uniform' over the complete space: the same threshold is applied for all face pairs. In our experiments we consider two ways of improving the graph construction: (i) using a k-nearest neighbourhood (kNN) definition, and (ii) differentiating between neighbours using real-valued weights.

It is important that there is enough difference between the edge weights of similar and dissimilar faces: direct use of similarity or distance values as edge weights tends to give poor results, and some non-linear transformation of these values using neighbourhood definition is crucial. The kNN method assigns non-zero edge weights between each face and its k most similar faces, which may be interpreted as an adaptive threshold, and leads to asymmetric connectivity.

If we set the edge weight w_{ij} for all j among the k nearest neighbors of node i to one, the densest component of the graph is the entire set of nodes. Clearly $f(V) = k$, and excluding any set of faces from V will remove at least k links per face and therefore $f(S) \leq f(V)$ for $S \subseteq V$. The constrained search introduced above, that includes at most one face from each image in S, does not suffer from this problem since it limits the number of faces and is a prerequisite for obtaining satisfactory results using kNN graphs.

In addition to using binary weights, we consider a linear kNN approach in which the closest face is assigned a weight k, the second $k-1$ and so on down to 1 for the k-th nearest and 0 for all other faces. This encodes the relative importance of the neighbors into our densest component search.

2.3. Feature Spaces and Similarity Measures

The methods described so far can be applied to any kind of feature space and associated similarity measure or distance measure. In the baseline method the distance between two faces is defined as the average distance between SIFT features of matched interest points. Interest points are detected using the Difference of Gaussians method [12] and represented using SIFT descriptors. Examples of detected interest points can be found in Figure 2. Two interest points, I_i^1 from face 1 and I_j^2 from face 2, match when the following criteria holds: (i) I_i^1 is most similar to I_j^2 among the detections on face 2, *i.e.* $\forall_{k \neq j}: \; d(I_i^1, I_k^2) > d(I_i^1, I_j^2)$, and vice-versa, (ii) the Euclidean distance between I_i^1 and I_j^2, both represented by coordinates normalized with respect to the bounding box of the face detection, is below a certain limit, to work as a geometrical constraint. We refer to this method for creating a face description using interest points as 'IP' in our experiments, and to the distance definition in terms of the average distance as 'AV'.

This distance measure ignores the number of matches between two faces, which is problematic: two faces with only one single very good match yield a lower distance than two faces with 10 slightly worse matching interest points. Motivated by established methods for object recognition, see e.g.[16], we propose using the number of matches between two faces as a measure for their similarity. The similarity measure based on counting the number of matches is referred to as 'CT' in our experiments.

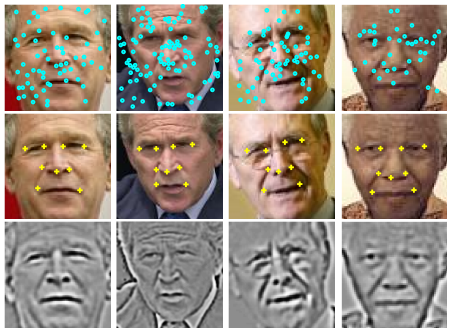


Figure 2. Example detections of interest points (IP, *top*), and facial features (FF, *middle*). The *bottom row* shows the normalised faces.

We can avoid the geometrical matching problem altogether by using a representation based on a fixed set of predefined locations in the face. We use a method that detects nine facial features [9], illustrated in Figure 2. The nine detected features plus an additional four (placed at the middle of both eyes, the middle of the mouth and between the eyes) are again represented using SIFT descriptors, which leads to 1664-dimensional face descriptors. The descriptors are extracted after image normalization which compensates for low-frequency lighting variations and suppresses noise with a Difference of Gaussians filter. This technique has recently been shown to lead to state-of-the-art performance on face recognition tasks [17] and improves the performance of our descriptors. In this feature space we can either use the average Euclidean distance over the 13 SIFT descriptor pairs as a distance measure, or count the number of matches for these 13 features using the criterion given above; the geometrical constraint then simply requires that for a matched pair I_i^1, I_j^2 we must have $i = j$. We use 'FF' to refer to results obtained using the facial feature detector.

3. Multi-person Naming

In this section we consider naming all faces in a database of captioned news images. For each face we want to know to which name in the caption it corresponds, or possibly that it corresponds to none of them: a *null* assignment. In this setting, we can use the following constraints: (i) a face can be assigned to at most one name, (ii) this name must appear in the caption, and (iii) a name can be assigned to at most one face. As an illustration, the seven admissible assignments for a document with two names and two faces

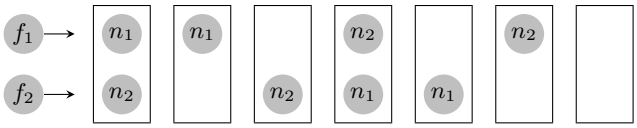


Figure 3. Illustration of the seven admissible assignments in a documents with two faces f_1 and f_2 and two names n_1 and n_2.

is given in Figure 3.

We extend the graph-based approach by trying to find subgraphs S_n of the similarity graph for each name n. This can be thought of as querying simultaneously for each name using the method for single-person search, which would comply with (ii) and (iii). But doing so in a straightforward manner could result in overlapping subgraphs, which would violate (i). Considering only the admissible assignments when concurrently searching for the set $\{S_n\}$ of subgraphs overcomes this problem as detailled in Section 3.1.

Because the number of admissible assignments for documents with more than a few names and faces quickly becomes intractable, we propose in Section 3.2 an efficient method for finding the best assignment based on a max-flow formulation that avoids explicitly considering all assignments. In Section 3.3 we describe our baseline for the multi-person experiments: the constrained mixture model approach of [3].

3.1. Similarity Graph Approach to Clustering

In the single-person query task, we search for the densest subgraph S of the similarity graph G implied by the text-based search. We extend this as follows: the similarity graph G is now computed considering all faces in the dataset. In this graph, we search simultaneously for subgraphs S_n corresponding to names n that are extracted automatically from the text using the named entity detector.

It is worth noting first that the number of example faces for different people varies greatly, from just one or two to hundreds or thousands. As a result, optimising the sum of the densities of subgraphs S_n leads to very poor results (we do include them in our experimental results for reference). Using the sum of the densities tends to assign an equal number of faces to each name, as far as allowed by the constraints, and therefore does not work well for very frequent and rare people. Instead we propose maximising the sum of edge weights within each subgraph:

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

Note that when $w_{ii} = 0$ this criterion does not differentiate between empty clusters and clusters with a single face. To avoid clusters with a single associated face, for which there are no other faces to corroborate the correctness of the assignment, we set w_{ii} to small negative values.

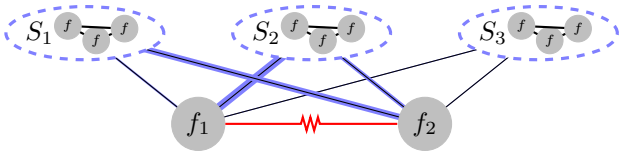


Figure 4. Example of a document with faces f_1 and f_2, and three names corresponding to subgraphs S_1, S_2 and S_3. Given the sum of edge weights (represented by width) that connect each face to the clusters, we search for the best admissible assignment.

Then, as stated, we cannot simply query independently for each name, as it would result in overlapping clusters. Instead, the subgraphs S_n can be obtained concurrently by directly maximizing eq. (2), while preserving the document constraints. Finding the optimal global assignment is computationally intractable, and we thus resort to approximate methods. As in the single-person case, the subgraphs associated with the names are initialized with all nodes where the name could be assigned. Then we iterate over documents and optimise eq. (2) per document. The iteration continues until a fixed-point is reached, which takes in practice 4 to 10 iterations. An illustration of this document-level objective maximisation is shown in Figure 4. In the next section we show how the optimal assignment for a document can be found efficiently.

3.2. Document-level Objective Optimisation

The number of admissible assignments for a document with F faces and N names is $\sum_{p=0}^{\min(F,N)} p!\binom{F}{p}\binom{N}{p}$, and thus quickly becomes impractically large. For instance, our fully-labeled data set contains documents with $F = 6$ faces and $N = 7$ names, yielding 37633 admissible assignments. Notably, the five largest documents amount for more than 90% of the number of admissible assignments to be evaluated over the full dataset.

Given the fact that assignments share many common sub-assignments – the underlying structure is a lattice – a large efficiency gain can be expected by not re-evaluating the shared sub-assignments. We therefore introduce a reduction of the optimisation problem to a well-studied minimum cost matching in a weighted bipartite graph [7]. This modelling takes advantage of this underlying structure and can be implemented efficiently. Its use is limited to objectives that can be written as a sum of 'costs' $c(f, n)$ for assigning face f to name n. The corresponding graphical representation is shown in Figure 5.

The names and faces problem differs from standard matching because we have to take into account *null* assignments, and this *null* value can be taken by any number of faces in a document. This is handled by having as many *null* nodes as there are faces and names. A face f can be paired with any name or its own copy of *null*, which is

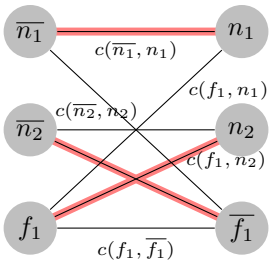


Figure 5. Example of the weighted bipartite graph corresponding to a document with one face and two names, for clarity only non-zero costs are shown. An example of a matching solution is given with the highlighted lines, it is interpreted as assigning face f_1 to name n_2 and not assigning name n_1.

written $\overline{f}$, and reciprocally, a name n can be paired with any face or its own copy of *null*, written $\overline{n}$. A pairing between f and n will require the pairing of $\overline{n}$ and $\overline{f}$ because of document constraints. The weights of the pairings are simply the costs of assigning a face f_i to the subgraph S_n, *i.e.* $-\sum_{f_j \in S_n} w_{ij}$, or to *null*. A bipartite matching problem is a specialized form of min-cost max-flow problem where all capacities equal one, and a min-cost max-flow problem is a special case of a linear programming problem efficiently solved using the simplex algorithm. It is then straightforward to obtain the minimum cost and the corresponding assignment from the pairing of nodes in the max-flow graph.

In Figure 6 we show how the processing time grows as a function of the product of the number of faces and names for the min-cost max-flow algorithm compared to a 'brute-force' loop over all admissible assignments. Clearly, the max-flow algorithm scales much better for large documents.

3.3. A Constrained Mixture Model Approach

In order to compare to previous work on naming faces in news images [3], we have implemented a constrained mixture model approach. We associate a Gaussian density in the feature space with each name, and an additional Gaussian is associated with *null*. The parameters of the latter will be fixed to the mean and variance of the ensemble of all faces in the data set, while the former will be estimated from the data. We assume a uniform distribution over the admissible assignments, and given the assignment we assume the features of each face f_i to be independently generated from the associated Gaussian. Given an assignment γ the likelihood of a document containing F faces is thus given by

$$p(\{f_1, \ldots, f_F\}|\gamma) = \prod_{i=1}^{F} p(f_i|\gamma), \qquad (3)$$

where $p(f_i|\gamma) = \mathcal{N}(f_i; \mu_n, \Sigma_n)$ and n is the name given by the assignment $(f_i, n) \in \gamma$. The sum of the log-likelihood of all documents is maximised using an EM algorithm that updates the parameters μ_n and Σ_n in the M-step, based on the assignments found in the E-step. By limiting the E-step to find the a posteriori maximum likelihood assignment we can again use the min-cost max-flow algorithm; here the costs $c(f, n)$ correspond to $-\ln \mathcal{N}(f; \mu_n, \Sigma_n)$.

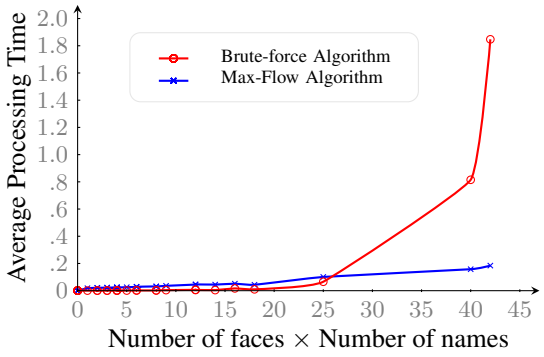


Figure 6. Average processing time of the brute force algorithm and the min-cost max-flow algorithm.

The generative model in [3] incorporates more information from the caption, but by leaving this out both methods use the same information and we can compare directly with our graph-based method. More caption features can be incorporated in the graph-based method by introducing additional terms that favor names of people who are likely to appear in the image based on textual analysis, although we did not explore this in our current work. In [11] the caption is treated as a bag-of-words using a variant of latent Dirichlet allocation [5]. However, their main focus was to obtain people-specific distributions over words; for face-naming the model was reported to perform worse than that of [3].

The generative model also applies to the single-person task, in which case we have just two mixture components: one represents the queried person, and the other is again fixed and fitted to all the faces returned on the query. In the single-person case the number of admissible assignments is just $F + 1$ for a document with F faces (choose one of the faces, or none), and therefore in the E-step the a posteriori probability for all assignments is readily computed.

4. Experiments and Results

In this section we first present experimental results for single-person retrieval, and then for multi-person naming.

4.1. Single-person Retrieval

To evaluate the performance of our modifications of the baseline method, we have selected the same 23 name queries as [14]. However, the actual data sets differ in the selection of documents and their ground truth. In our case, the documents are selected purely by a text search of the captions, and the ground truth is hand-labelled, whereas in [14] the method of Berg *et al.* [3] was used to select the documents and to produce the ground truth, *i.e.* an imperfect ground truth was used. Figure 7 (black line) shows, for each name, the ratio of correct faces in the returned set, *i.e.* the precision of the text-based query.

Recall	GR	DC	DC+LS	Ozkan [14]
75	68.2	70.6	71.3	69.3
85	62.8	63.5	66.1	65.2

Table 1. Average precision for two levels of recall and for different search algorithms, using IP-AV features and an ϵ-neighborhood.

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

Table 2. Average precision for different features, similarity measures, and recall levels, using graph-based methods with DC+LS.

To measure performance we use precision and recall. The precision is the percentage of faces corresponding to the queried person with respect to all returned faces. The recall is the percentage of correct faces in the set of all returned faces.

Document Constrained Search. We start by comparing the performance of the three different search techniques described in Section 2.1: the baseline greedy search (GR), document constrained search (DC), and document constrained search with local search (DC+LS). The results are presented in Table 1. With our implementation of the greedy search method we obtained an average precision of 68.2% for a recall of 75%. These results are slightly different from those in [14], where an average precision of 69.3% for the same recall is reported. This can be explained by the difference in the dataset and labeling. The results clearly show that the document constrained search improves the baseline method, and that adding local search leads to a further improvement. The results given here are for IP-AV using an ϵ-neighborhood; the same trend is observed for different features, similarities and graph construction methods.

Similarity Graphs & Feature Spaces. We compare the different methods for graph construction, the different feature spaces and similarity measures. To construct the graph we consider the ϵ-neighborhood method and the linear kNN method, where we set k to a percentage of the number of faces in the return set to allow for a suitable k for different sizes of the return set. We consider interest points (IP) and facial features (FF), and use these to compute either the average distance between SIFT descriptors (AV), or the number of matches (CT). We summarize the performance for the different combinations of these methods in Table 2.

For both neighborhood methods, the best performance is obtained when using the count of matched interest points (IP-CT). The results show that match count (CT) always outperforms the corresponding average distance between matched points (AV), and that, when using CT, kNN is always preferable to an ϵ-neighborhood.

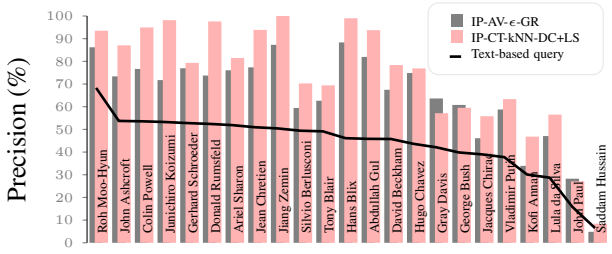


Figure 7. Comparison of precision at 85% recall for individual queries using the baseline method (IP-AV-ϵ-GR), and our best method (IP-CT-kNN-DC+LS). The queries were sorted by the precision of the text-based result.

In Figure 7 we show the precision for individual queries using the baseline method (IP-AV-ϵ-GR) and our best performing method (IP-CT-kNN-DC+LS) for 85% recall. For almost all queries we outperform the baseline significantly, *i.e.*, up to 26.4%. The figure also shows significant difference in results among queries, due to smaller or larger variations in appearance. In Figure 8 we show the 25 top-ranked faces returned on two queries (the examples were chosen to illustrate good and bad performance); faces were ranked based on their contribution to the density f in eq. (1), *i.e.* their weighted *degree* in the graph.

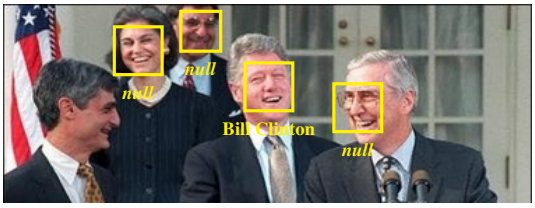
Generative Model. We also compare the graph-based method with the generative model described in Section 3.3. This generative model can only be applied when using the facial feature detector (FF), as a fixed size representation is required. The model does not have any parameters to adjust and obtains an average precision of 75.5% with an average recall of 67.8%. The graph-based method closest to the generative model (FF-AV-kNN-DC+LS) has 76.7% recall for the same precision. Our best method (IP-CT-kNN-DC+LS) obtains 78.7% recall, *i.e.*, an improvement of 11% in recall.

To summarize, the proposed modifications outperform the baseline method [14] with an increase of around 10% in precision for a recall of 85%. Similar results hold when comparing to the generative approach.

4.2. Multi-person Naming

Proper evaluation of the multi-person naming task requires a full ground-truth labeling for every face in the data set. On the full Yahoo! News data set, this information is not available. Therefore, we annotated a smaller data set from Yahoo! News. It contains 857 documents, with a total of 1183 detected faces and 1528 named entities (424 unique names). Due to the smaller number of samples for each person, and the fact that there are on average more faces and more names in each document, we expect performance to be higher when using the bigger data set. Because of the relatively small data set our algorithms sometimes does not

Figure 8. The top 25 faces retrieved for a well-performing example query (David Beckham, top) and a poor one (Pope John Paul, bottom); Faces are ranked from left based on their degree in the subgraph. Faces that do not correspond to the queried person are marked with a red bar. Pope John Paul appears only rarely in documents where he is mentionned, hence making the query a very difficult task.



Lloyd Bentsen is pictured here announcing his retirement in 1994 at the White House with former US President **Bill Clinton**, Chief of Staff **Leon Panetta**, **Robert Rubin** and **Judy Rubin** (...)

Figure 9. Example where our FF-CT-LT multi-person naming assigns only one name and three times *null*. This is due to the fact that except for Bill Clinton the faces appear only once in our dataset and do not allow to corroborate any additional assignment.

manage to name all faces, see Figure 9. Nevertheless, a comparison between different methods can be carried out. See Figure 1 for examples of successful face naming using FF-CT-LT (see below).

We used two measures for performance. We refer to the percentage of correct assignments for all the faces, including assignments to *null*, as the overall accuracy given for the best set of parameters as in [3]. Our second measure is the percentage of correct assignments among the faces that were assigned to a name, refered to as precision. Note that the best overall accuracy corresponds to one point on the precision/number-of-named-faces curve, see Figure 10.

The accuracy of the different assignment methods is summarised in Table 3. The generative baseline approach (Gen) using the facial features vector descriptor (FF) achieves 48.7% accuracy. A gain can be obtained by preprocessing the data (GenPP) following [3] with Principal Components Analysis (PCA) and Fisher's Linear Discriminant Analysis (LDA). LDA uses noisy supervised information gathered from documents with only one name and one face to find the directions of maximum variance between groups of faces belonging to different people while minimising the variance within these groups. These two preprocessing steps improve performance to 51.7%. For reference, the accuracy when maximising densities (DENS) for the subgraphs associated with the different names, using the FF-CT similarity measure is 40.1%. All our graph based methods described in Section 3 significantly outperform the generative and DENS approaches, the best one obtaining 63.5% overall accuracy.

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

Table 3. Overall accuracy (%) for different face naming algorithms. Values for generative methods are the average over 15 runs with different random initializations.

We also compared the different similarity measures, *i.e.* the number of matching facial features (FF-CT), interest point matching (IP-CT), and average distance between facial features (FF-AV). The FF-AV method is most closely related to the generative baseline method, and using this method we obtain 59.3% accuracy which represents a 7.6% improvement over the GenPP baseline. We observed even larger improvements when using match count (CT) in combination with facial features (FF) or interest points (IP): 63.5% and 62.0% respectively.

We compared the different ways to construct graphs from similarities: Hard Thresholding (HT), where we obtain a binary graph after keeping only the edges indicating sufficiently high similarity, Linear Thresholding (LT), where similarities above threshold t are linearly transformed using the formula $w' = \max(w - t, 0)$, and linear k-Nearest Neighbours (kNN, see Section 2.2), where k is chosen at the document-level in proportion to the number of faces that could be assigned to a name appearing in the document. The accuracy using HT is 63.4%, and LT performs slightly better at 63.5%. Both methods perform significantly better than kNN (59.1%). This difference from the results on the single person task, where kNN performed better, can be explained by the small size of our fully annotated data set, in which many faces should not be linked to any other.

Figure 10 shows the precision vs. the number of named faces for some of the considered methods. For IP-CT-LT and FF-CT-LT, the parameter varied is the value of LT. For the generative methods, it is the threshold on the Mahalanobis distances to the cluster centers. Curves are averaged over 15 randomly-initialized runs. All curves start on the right, where no thresholding is performed and most faces are named. Also shown are the points of each curve where the overall accuracy is obtained. These results show that Gaussians are poor estimates of the cluster densities in the feature space, while the graph-based approach is more flexible and obtains excellent precision for lower recall levels.

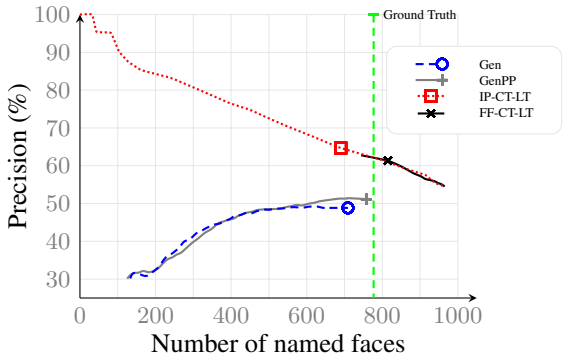


Figure 10. Precision vs. number of named faces for some of our naming algorithms. Refer to the text for details.

5. Conclusions

We considered graph-based and generative approaches to solving two tasks: finding faces of a single person, and naming all the faces in a data set. We have shown that we can obtain significant improvements over existing methods by improving and extending an existing graph-based method.

For single-person retrieval, improvements were obtained by (i) constraining the return set to include at most one face from each image, (ii) using k-nearest neighborhood graphs rather than an ϵ-neighborhood, and (iii) using more reliable measures of similarity between faces. Averaging over 23 queried names we obtain an average precision of 77.6% (resp. 73.0%) for a recall of 75% (resp. 85%), *i.e.*, improvements of more than 9% compared to the baseline method.

We extended the graph-based approach to multi-person naming, which raised a complexity issue that we addressed by proposing an efficient method based on min-cost max-flow graphs to find the optimal name-face assignment in a single document under unique matching constraints. This method leads to dramatic computational savings when the number of names and faces is relatively large, and applies for both the graph-based method and the generative model. Our novel method was shown to outperform a generative approach that was previously proposed for this task. In this setting we obtained 63.5% of overall correct assignments of faces, to names in the caption or *null*, and among the faces assigned to names our method scores 61.5% correct. Compared to the generative baseline method (GenPP) these are improvements of 11.8% and 11.3% respectively.

Adding a language model, and enhancing face detection and facial feature localization can bring further improvements, as this will lead to cleaner data sets from which to construct the similarity graphs. The potential applications of the methods proposed in this paper include web-based photo retrieval by name, automatic photo annotation, and news digest applications. Our methods are general enough to be used for other visual retrieval tasks among images with weak forms of annotation such as captions, we will explore such possibilities in future work.

Acknowledgements

We would like to thank Tamara Berg for sharing the data set with us. This work was supported by the European funded research project CLASS.

References

[1] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[2] R. Bekkerman and J. Jeon. Multi-modal clustering for multimedia collections. In *CVPR*, 2007.

[3] T. Berg, A. Berg, J. Edwards, M. Maire, R. White, Y. Teh, E. Learned-Miller, and D. Forsyth. Names and faces in the news. In *CVPR*, pages 848–854, 2004.

[4] T. L. Berg and D. A. Forsyth. Animals on the web. In *CVPR*, pages 1463–1470, 2006.

[5] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[6] M. Charikar. Greedy approximation algorithms for finding dense components in a graph. In *Proceedings of APPROX*, pages 139–152, 2000.

[7] T. Cormen. *Introduction to algorithms*. MIT Press, 2001.

[8] K. Deschacht and M. Moens. Efficient hierarchical entity classification using conditional random fields. In *Workshop on Ontology Learning and Population*, 2006.

[9] M. Everingham, J. Sivic, and A. Zisserman. 'Hello! My name is... Buffy' - automatic naming of characters in TV video. In *BMVC*, pages 889–908, 2006.

[10] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from Google's image search. In *ICCV*, pages 1816–1823, 2005.

[11] V. Jain, E. Learned-Miller, and A. McCallum. People-LDA: Anchoring topics to people using face recognition. In *ICCV*, 2007.

[12] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

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

[14] D. Ozkan and P. Duygulu. A graph based approach for naming faces in news photos. In *CVPR*, pages 1477–1482, 2006.

[15] S. Satoh and T. Kanade. Name-It: association of face and name in video. In *CVPR*, pages 368–373, 1997.

[16] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. In *IEEE Trans. on PAMI*, pages 530–535, 1997.

[17] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. In *Analysis and Modelling of Faces and Gestures*, volume 4778 of *LNCS*, pages 168–182. Springer, 2007.

[18] J. Verbeek and B. Triggs. Region classification with Markov field aspect models. In *CVPR*, 2007.