Mining Weakly Labeled Web Facial Images for Search-Based Face Annotation

Dayong Wang[†], Steven C. H. Hoi[†], Ying He[†], Jianke Zhu[‡]
[†]School of Computer Engineering, Nanyang Technological University, Singapore
[‡]College of Computer Science, Zhejiang University, Hangzhou, 310027, P.R. China
E-mail:{s090023, chhoi, yhe}@ntu.edu.sg, jkzhu@zju.edu.cn

Abstract—This paper investigates a framework of Search-Based Face Annotation (SBFA) by mining weakly labeled facial images that are freely available on the World Wide Web (WWW). One challenging problem for search-based face annotation scheme is how to effectively perform annotation by exploiting the list of most similar facial images and their weak labels that are often noisy and incomplete. To tackle this problem, we propose an effective Unsupervised Label Refinement (ULR) approach for refining the labels of web facial images using machine learning techniques. We formulate the learning problem as a convex optimization and develop effective optimization algorithms to solve the large-scale learning task efficiently. To further speed up the proposed scheme, we also propose a clustering-based approximation algorithm which can improve the scalability considerably. We have conducted an extensive set of empirical studies on a large-scale web facial image testbed, in which encouraging results showed that the proposed ULR algorithms can significantly boost the performance of the promising SBFA scheme.

Index Terms—face annotation, content-based image retrieval, machine learning, label refinement, web facial images, weak label

1 Introduction

Due to the popularity of various digital cameras and the rapid growth of social media tools for internet-based photo sharing [1], recent years have witnessed an explosion of the number of digital photos captured and stored by consumers. A large portion of photos shared by users on the Internet are human facial images. Some of these facial images are tagged with names, but many of them are not tagged properly. This has motivated the study of *auto face annotation*, an important technique that aims to annotate facial images automatically.

Auto face annotation can be beneficial to many real-world applications. For example, with auto face annotation techniques, online photo-sharing sites (e.g., Facebook) can automatically annotate users' uploaded photos to facilitate online photo search and management. Besides, face annotation can also be applied in news video domain to detect important persons appeared in the videos to facilitate news video retrieval and summarization tasks [2], [3].

Classical face annotation approaches are often treated as an extended face recognition problem, where different classification models are trained from a collection of well-labeled facial images by employing the supervised or semi-supervised machine learning techniques [2], [4]–[7]. However, the "model-based face annotation" techniques are limited in several aspects. First, it is usually time-consuming and expensive to collect a large amount of human-labeled training facial images. Second, it is usually difficult to generalize the models when new training data or new persons are added, in which an intensive re-training process is usually required. Last but not least, the annotation/recognition performance often scales poorly when the number of persons/classes is very large.

Recently, some emerging studies have attempted to explore a promising search-based annotation paradigm for facial image annotation by mining the World Wide Web (WWW), where a massive number of weakly labeled facial images are freely available. Instead of training explicit classification models by the regular model-based face annotation approaches, the search-based face annotation (SBFA) paradigm aims to tackle the automated face annotation task by exploiting content-based image retrieval (CBIR) techniques [8], [9] in mining massive weakly labeled facial images on the web. The SBFA framework is data-driven and model-free, which to some extent is inspired by the search-based image annotation techniques [10]–[12] for generic image annotations. The main objective of SBFA is to assign correct name labels to a given query facial image. In particular, given a novel facial image for annotation, we first retrieve a short list of top K most similar facial images from a weakly labeled facial image database, and then annotate the facial image by performing voting on the labels associated with the top K similar facial images.

One challenge faced by such SBFA paradigm is how to effectively exploit the short list of candidate facial images and their weak labels for the face name annotation task. To tackle the above problem, we investigate and develop a search-based face annotation scheme by focusing on tackling this problem. In particular, we propose a novel unsupervised label refinement scheme by exploring machine learning techniques to enhance the labels purely from the weakly labeled data without human manual efforts. We also propose a Clustering-based Approximation (CBA) algorithm to improve the efficiency and scalability. As a summary, the main contributions of this paper include the following:

- We investigate and implement a promising search-based face annotation scheme by mining large amount of

weakly labeled facial images freely available on the WWW.
- We propose a novel Unsupervised Label Refinement (ULR) scheme for enhancing label quality via a graph-based and low-rank learning approach.
- We propose an efficient clustering based approximation (CBA) algorithm for large scale label refinement problem.
- We conducted an extensive set of experiments, in which encouraging results were obtained.

We note that a short version of this work had appeared in SIGIR2011 [13]. This journal article has been significantly extended by including a substantial amount of new content. The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 gives an overview of the proposed search-based face annotation framework. Section 4 presents the proposed unsupervised label refinement scheme. Section 5 shows our experimental results of performance evaluation, and Section 6 discusses the limitation of our work. Finally, Section 7 concludes this paper.

2 Related Work

Our work is closely related to several groups of research work.

The first group of related work is on the topics of face recognition and verification, which are classical research problems in computer vision and pattern recognition and have been extensively studied for many years [14], [15]. Recent years have observed some emerging benchmark studies of unconstrained face detection and verification techniques on facial images that are collected from the web, such as the LFW benchmark studies [16]–[19]. Some recent study had also attempted to extend classical face recognition techniques for face annotation tasks [7]. Comprehensive reviews on face recognition and verification topics can be found in some survey papers [15], [20], [21] and books [22], [23].

The second group is about the studies of generic image annotation [24]–[27]. The classical image annotation approaches [28]–[30] usually apply some existing object recognition techniques to train classification models from human-labeled training images or attempt to infer the correlation/probabilities between images and annotated keywords. Given limited training data, semi-supervised learning methods have also been used for image annotation [31]–[33]. For example, Wang et al. [31] proposed to refine the model-based annotation results with a label similarity graph by following random walk principle [34]. Similarly, Pham et al. [32] proposed to annotate unlabeled facial images in video frames with an iterative label propagation scheme. Although semi-supervised learning approaches could leverage both labeled and unlabeled data, it remains fairly time-consuming and expensive to collect enough well-labeled training data in order to achieve good performance in large-scale scenarios. Recently, the search-based image annotation paradigm has attracted more and more attention [10], [35], [36]. For example, Russell et al. [36] built a large collection of web images with ground truth labels to facilitate object recognition research. However, most of these works were focused on the indexing, search, and feature extraction techniques. Unlike these existing works, we propose a novel unsupervised label refinement scheme that is focused on optimizing the label quality of facial images towards the search-based face annotation task.

The third group is about face annotation on personal/family/social photos. Several studies [37]–[40] have mainly focused on the annotation task on personal photos, which often contain rich contextual clues, such as personal/family names, social context, geo-tags, timestamps, etc. The number of persons/classess is usually quite small, making such annotation tasks less challenging. These techniques usually achieve fairly accurate annotation results, in which some techniques have been successfully deployed in commercial applications, e.g., Apple iPhoto, Google Picasa, Microsoft easyAlbum [38], and Facebook face auto-tagging solution.

The fourth group is about the studies of face annotation in mining weakly labeled facial images on the web. Some studies consider a human name as the input query, and mainly aim to refine the text-based search results by exploiting visual consistency of facial images. For example, Ozkan and Duygulu [41] proposed a graph-based model for finding the densest sub-graph as the most related result. Following the graph-based approach, Le and Satoh [42] proposed a new local density score to represent the importance of each returned images, and Guillaumin et al. [43] introduced a modification to incorporate the constraint that a face is only depicted once in an image. On the other hand, the generative approach like the gaussian mixture model was also been adopted to the name-based search scheme [5], [43] and achieved comparable results. Recently, a discriminant approach was proposed in [44] to improve over the generative approach and avoid the explicit computation in graph-based approach. By using ideas from query expansion [45], the performance of name-based scheme can be further improved with introducing the images of the "friends" of the query name. Unlike these studies of filtering the text-based retrieval results, some studies have attempted to directly annotate each facial image with the names extracted from its caption information. For example, Berg et al. [46] proposed a possibility model combined with a clustering algorithm to estimate the relationship between the facial images and the names in their captions. For the facial images and the detected names in the same document(a web image and its corresponding caption), Guillaumin et al. [43] proposed to iteratively update the assignment based on a minimum cost matching algorithm. In their follow-up work [44], they further improve the annotation performance by using distance metric learning techniques to achieve more discriminative feature in low-dimension space.

Our work is different from the above previous works in two main aspects. First of all, our work aims to solve the general content-based face annotation problem using the search-based paradigm, where facial images are directly used as query images and the task is to return the corresponding names of the query images. Very limited research progress has been reported on this topic. Some recent work [47] mainly addressed the face retrieval problem, in which an effective image representation has been proposed using both local and global features. Secondly, based on initial weak labels, the proposed Unsupervised Label Refinement (ULR) algorithm learns an enhanced new label matrix for all the facial images in the whole name space; however, the caption-based annotation scheme only considers the assignment between the facial images and the names appeared in their corresponding surrounding-text. As a result, the caption-based annotation scheme is only applicable to the scenario where both images and their captions are available, and cannot be applied to our SBFA framework due to the lack of complete caption information.

The fifth group is about the studies of purifying web facial images, which aims to leverage noisy web facial images for face recognition applications [5], [48]. Usually these works are proposed as a simple preprocessing step in the whole system without adopting sophisticated techniques. For example, the work in [5] applied a modified k-means clustering approach for cleaning up the noisy web facial images. Zhao et al. [48] proposed a consistency learning method to train face models for the celebrity by mining the text-image co-occurrence on the web as a weak signal of relevance towards supervised face learning task from a large and noisy training set. Unlike the above existing works, we employ the unsupervised machine learning techniques and propose a graph-based label refinement algorithm to optimize the label quality over the whole retrieval database in the SBFA task.

Finally, we note that our work is also related to our recent work of the WLRLCC method in [49] and our latest work on the unified learning scheme in [50] [1]. Instead of enhancing the label matrix over the entire facial image database, the WLRLCC algorithm [49] is focused on learning more discriminative features for the top retrieved facial images for each individual query, which thus is very different from the ULR task in this paper. Last but not least, we note that the learning methodology for solving the unsupervised label refinement task are partially inspired by some existing studies in machine learning, including graph-based semi-supervised learning and multi-label learning techniques [51]–[53].

3 Search-Based Face Annotation

Fig. 1 illustrates the system flow of the proposed framework of Search-Based Face Annotation (SBFA), which consists of the following steps: (1) facial image data collection; (2) face detection and facial feature extraction; (3) high-dimensional facial feature indexing; (4) learning to refine weakly labeled data; (5) similar face retrieval; (6) face annotation by majority voting on the similar faces with the refined labels. The first four steps are usually conducted before the test phase of a face annotation task, while the last two steps are conducted during the test phase of a face annotation task, which usually should be done very efficiently. We briefly describe each step below.

The first step is the data collection of facial images as shown in Fig. 1(a), in which we crawled a collection of facial images from the WWW by an existing web search engine (i.e., Google) according to a name list that contains the names of persons to be collected. As the output of this crawling process, we shall obtain a collection of facial images, each of them is associated with some human names. Given the nature of web images, these facial images are often noisy, which do not always correspond to the right human name. Thus, we call such kind of web facial images with noisy names as weakly labeled facial image data.

The second step is to pre-process web facial images to extract face-related information, including face detection and alignment, facial region extraction, and facial feature representation. For face detection and alignment, we adopt the unsupervised face alignment technique proposed in [54]. For facial feature representation, we extract the GIST texture features [55] to represent the extracted faces. As a result, each face can be represented by a d-dimensional feature vector.

The third step is to index the extracted features of the faces by applying some efficient high-dimensional indexing technique to facilitate the task of similar face retrieval in the subsequent step. In our approach, we adopt the Locality-Sensitive Hashing (LSH) [56], a very popular and effective high-dimensional indexing technique.

Besides the indexing step, another key step of the framework is to engage an unsupervised learning scheme to enhance the label quality of the weakly labeled facial images. This process is very important to the entire search-based annotation framework since the label quality plays a critical factor in the final annotation performance.

All the above are the processes before annotating a query facial image. Next we describe the process of face annotation during the test phase. In particular, given a query facial image for annotation, we first conduct a similar face retrieval process to search for a subset of most similar faces (typically top K similar face examples) from the previously indexed facial database. With the set of top K similar face examples retrieved from the database, the next step is to annotate the facial image with a label (or a subset of labels) by employing a majority voting approach that combines the set of labels associated with these top K similar face examples.

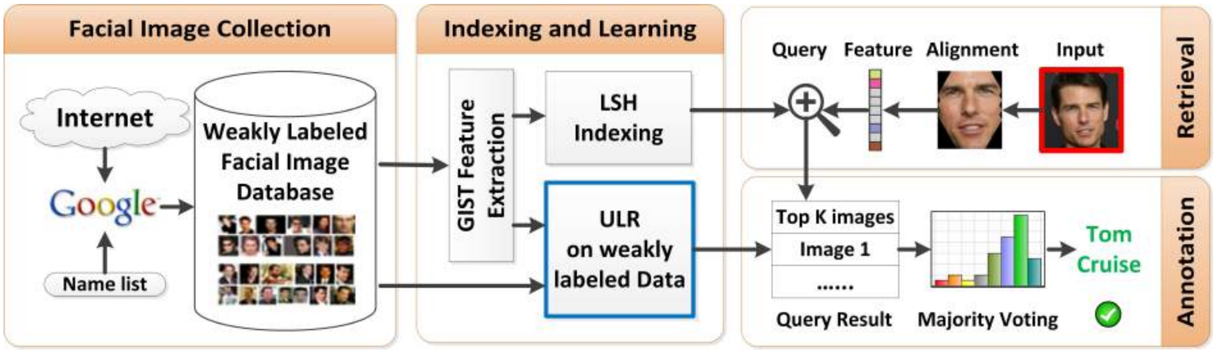
In this paper, we focus our attention on one key step of the above framework, i.e., the unsupervised learning process to refine labels of the weakly labeled facial images.

4 Unsupervised Label Refinement by Learning on Weakly Labeled Data

4.1 Preliminaries

We denote by $X \in \mathbb{R}^{n \times d}$ the extracted facial image features, where n and d represent the number of facial images and the number of feature dimensions, respectively. Further we denote by $\Omega = \{n_1, n_2, \dots, n_m\}$ the list of human names for annotation, where m is the total number of human names. We also denote by $Y \in [0,1]^{n \times m}$ the initial raw label matrix to describe the weak label information, in which the i-th row Y_{i*} represents the label vector of the i-th facial image $\mathbf{x}_i \in \mathbb{R}^d$. In our application, Y is often noisy and incomplete. In particular, for each weak label value Y_{ij}, $Y_{ij} \neq 0$ indicates that the i-th facial image $\mathbf{x}_i$ has the label name n_j, while $Y_{ij} = 0$ indicates that the relationship between i-th facial image $\mathbf{x}_i$ and j-th name is unknown. Note that we usually have $\|Y_{i*}\|_0 = 1$ since each facial image in our database was uniquely collected by a single query.

[1] These two works were proposed and published after the conference version of this study [13].



(a) (b) (c)

Fig. 1. The system flow of the proposed search-based face annotation (SBFA) scheme. (a) We collect weakly labeled facial images from WWW using web search engines. (b) We pre-process the crawled web facial images, including face detection, face alignment, and feature extraction for the detected faces; after that, we apply LSH to index the extracted high-dimensional facial features. We apply the proposed Unsupervised Label Refinement (ULR) method to refine the raw weak labels together with the proposed Clustering-based Approximation algorithms for improving the scalability. (c) We search for the query facial image to retrieve the top K similar images and use their associated names for voting towards auto annotation.

Following the terminology of graph-based learning methodology, we build a sparse graph by computing the weight matrix $W = [W_{ij}] \in \mathbb{R}^{n\times n}$, where W_{ij} represents the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$.

4.2 Problem Formulation

The goal of the Unsupervised Label Refinement (ULR) problem is to learn a refined label matrix $F^* \in \mathbb{R}^{n\times m}$, which is expected to be more accurate than the initial raw label matrix Y. This is a challenging task since we have nothing else but the raw label matrix Y and the data examples X themselves. To tackle this problem, we propose a graph-based learning solution based on a key assumption of "label smoothness", i.e., the more similar the visual contents of two facial images, the more likely they share the same labels. The label smoothness principle can be formally formulated as an optimization problem of minimizing the following loss function $E_s(F,W)$:

$$E_s(F,W) = \frac{1}{2}\sum_{i,j=1}^{n} W_{ij}\|F_{i*} - F_{j*}\|_F^2 = tr(F^\top LF) \quad (1)$$

where$\|\cdot\|_F$ denotes the Frobenius norm, W is the weight matrix of a sparse graph constructed from the n facial images, $L = D - W$ denotes the Laplacian matrix where D is a diagonal matrix with the diagonal elements as $D_{ii} = \sum_{j=1}^{n} W_{ij}$, and tr denotes the trace function.

Directly optimizing the above loss function is problematic as it will yield a trivial solution. To overcome this issue, we notice that the initial raw label matrix usually, though being noisy, still contains some correct and useful label information. Thus, when we optimize to search for F, we shall avoid the solution F being deviated too much from Y. To this end, we formulate the following optimization task for the unsupervised label refinement by including a regularization term $E_p(F,Y)$ to reflect this concern:

$$F^* = \arg\min_{F\geq 0} E_s(F,W) + \alpha \cdot E_p(F,Y) \quad (2)$$

where α is a regularization parameter and $F \geq 0$ enforces F is nonnegative. Next we discuss how to define an appropriate function for $E_p(F,Y)$.

One possible choice of $E_p(F,Y)$ is to simply set $E_p(F,Y) = \|F - Y\|_F^2$. This is however not appropriate as Y is often very sparse, i.e., many elements of Y are zeros due to the incomplete nature of Y. Thus, the above choice is problematic since it may simply force many elements of F to zeros without considering the label smoothness. A more appropriate choice of the regularization should be applied only to those nonzero elements of Y. To this end, we propose the following choice of $E_p(F,Y)$:

$$E_p(F,Y) = \|(F - Y)\circ S\|_F^2 \quad (3)$$

where S is a "sign" matrix $S = [sign(Y_{ij})]$ where $sign(x) = 1$ if $x > 0$ and 0 otherwise, and $\circ$ denotes the Hadamard product (i.e., the entrywise product) between two matrices.

Finally, we notice that the solution of the optimization in (3) is generally dense, which is again not desired since the true label matrix is often sparse. To take the sparsity into consideration, we introduce a sparsity regularizer $E_e(F)$ by following the "exclusive lasso" technique [57]:

$$E_e(F) = \sum_{i=1}^{n}(\|F_{i*}\|_1)^2 \quad (4)$$

where we introduce an ℓ_1 norm to combine the label weights for the same person with respect to different names, and an ℓ_2 norm to combine the label weights of different persons together. Combining this regularizer and the previous formulation, we have the final formulation as follows:

$$\begin{aligned} F^* &= \arg\min_{F\geq 0} g(F) \quad (5)\\ g(F) &= E_s(F,W) + \alpha E_p(F,Y) + \beta E_e(F) \end{aligned}$$

where $\alpha \geq 0$ and $\beta \geq 0$ are two regularization parameters. The above formulation combines all the terms in the objective function, which we refer it to as "Soft-Regularization Formulation" or "SRF" for short.

Another way to introduce the sparsity is to formulate the optimization by including some convex sparsity constraints, which leads to the following formulation:

$$\begin{aligned} F^* \quad &= \quad \arg\min_{F\geq 0} E_s(F,W) + \alpha E_p(F,Y) \\ s.t. \quad & \|F_{i*}\|_1 \leq \varepsilon, i = 1,\ldots,n \end{aligned} \tag{6}$$

where $\alpha \geq 0$ and $\varepsilon > 1$. We refer to this formulation as "Convex-Constraint Formulation" or "CCF" for short.

It is not difficult to see that the above two formulations are convex, which thus can be solved with global optima by applying convex optimization techniques. Next, we discuss efficient algorithms to solve the above optimization tasks.

4.3 Algorithms

The above optimization tasks belong to convex optimization or more exactly quadratic programming (QP) problems. It seems to be possible to solve them directly by applying generic QP solvers. However, this would be computationally highly intensive since matrix F can be potentially very large, e.g. for a large 400-person database, the F is a huge matrix that consists of over 10 million variables, which is almost infeasible to be solved by any existing generic QP solver.

4.3.1 *Algorithm for Soft-Regularization Formulation*

We firstly adopt an efficient algorithm to solve the problem in Eq.5, then propose a coordinate descent based approach to improve the scalability. By vectorizing matrix $F \in \mathbb{R}^{n\times m}$ into a column vector $\tilde{\mathbf{f}} = vec(F) \in \mathbb{R}^{(n\cdot m)\times 1}$, we can reformulate $g(F)$ as follows:

$$\begin{aligned} g(F) &= tr(F^\top LF) + \alpha\|(F-Y)\circ S\|_F^2 + \beta\|F\cdot \mathbf{1}\|_F^2 \\ &= \tilde{\mathbf{f}}^\top Q\tilde{\mathbf{f}} + \mathbf{c}^\top\tilde{\mathbf{f}} + \mathbf{h} \end{aligned} \tag{7}$$

where $\circ$ denotes the Hadamard product, $\otimes$ denotes the Kronecker product, $\tilde{\mathbf{y}} = vec(Y)$, $\tilde{\mathbf{s}} = vec(S)$, $\mathbf{1}$ is all one column vector, $U = I_m \otimes L^\top$, $V = (\mathbf{1}^\top \otimes I_n)$, $R = diag(\tilde{\mathbf{s}})$, $Q = U + \alpha R + \beta V^\top V$, $\mathbf{c} = -2\alpha R^\top \tilde{\mathbf{y}}$, $\mathbf{h} = \alpha\tilde{\mathbf{y}}^\top R\tilde{\mathbf{y}}$ and I_k is an identity matrix with dimension $k \times k$.

As shown in the vectorizing result, the optimization is clearly a QP problem. To efficiently solve this problem, we propose an accelerated multi-step gradient algorithm, which converges at $O(\frac{1}{k^2})$, k is the iteration step.

First of all, we reformulate the QP problem as follows:

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} q(\mathbf{x}|Q,\mathbf{c}) = \mathbf{x}^\top Q\mathbf{x} + \mathbf{c}^\top\mathbf{x} \quad \textbf{s.t. } \mathbf{x} \geq 0 \tag{8}$$

We then define a linear approximation function $p_t(\mathbf{x},\mathbf{z})$ for the above function q at point $\mathbf{z}$:

$$p_t(\mathbf{x},\mathbf{z}) = q(\mathbf{z}) + <\mathbf{x}-\mathbf{z}, \nabla q(\mathbf{z})> + \frac{t}{2}\|\mathbf{x}-\mathbf{z}\|_F^2 \tag{9}$$

where t is the Lipshitz constant of ∇q. In order to achieve the optimal solution $\mathbf{x}^\star$, we will update two sequences $\{\mathbf{x}^{(k)}\}$ and $\{\mathbf{z}^{(k)}\}$, recursively. Commonly at each iteration k, the variance $\mathbf{z}^{(k)}$ is named as *search point* and used for combining the two previous approximate solutions $\mathbf{x}^{(k-1)}$ and $\mathbf{x}^{(k-2)}$. The approximation $\mathbf{x}^{(k)}$ is achieved by solving the following optimization:

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} \; p_t(\mathbf{x},\mathbf{z}^{(k)}) \quad \textbf{s.t. } \mathbf{x} \geq 0 \tag{10}$$

After ignoring terms that do not depend on $\mathbf{x}$, the former optimization problem Eq.10 could be equally presented as:

$$\min_{\mathbf{x}\geq\mathbf{0}} \mathbf{g}^\top\mathbf{x} + \frac{t}{2}\|\mathbf{x}-\mathbf{z}^{(k)}\|^2 = t\sum_i[\frac{1}{2}(x_i - z_i^{(k)})^2 + \frac{g_i}{t}x_i] \tag{11}$$

where $\mathbf{g} = 2Q\mathbf{z}^{(k)} + \mathbf{c}$. The solution could be shown directly as follows:

$$x_i = \max(z_i^{(k)} - g_i/t, 0) \tag{12}$$

Finally, Algorithm1 summarizes the optimization progress.

Algorithm 1: Multi-step Gradient Algorithm for ULR

```
Input: $Q \in \mathbb{R}^{(n\cdot m)\times(n\cdot m)}$, $\mathbf{c} \in \mathbb{R}^{n\cdot m}$, $t \in \mathbb{R}$
Output: $\mathbf{x}^\star$
begin
    $\alpha_0 = 1; k = 1$ ; $\mathbf{z}^{(0)} = \mathbf{x}^{(0)} = \mathbf{x}^{(-1)} = 0$;
    repeat
        Case SRF : Achieve $\mathbf{x}^{(k)}$ with Eq. (10);
        Case CCF : Achieve $\mathbf{x}^{(k)}$ with Eq. (15);
        $\alpha_k = \frac{1+\sqrt{4\alpha_{k-1}^2+1}}{2}$;
        $\mathbf{z}^{(k)} = \mathbf{x}^{(k)} + \frac{\alpha_{k-1}-1}{\alpha_k}(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})$;
        $k = k + 1$;
    until CONVERGENCE;
```

To further improve the scalability, we propose a coordinate descent approach to solving the optimization iteratively. This can take advantages of the power of parallel computation when solving a very large-scale problem.

For the proposed coordinate descent approach, at each iteration, we optimize only one label vector F_{i*} by leaving the other vectors $\{F_{j*}|j \neq i\}$ intact. Specifically, at the $(t+1)$-th iteration, we define the following optimization problem for updating $F_{i*}^{(t+1)}$ with $F^{(t)}$:

$$F_{i*}^{(t+1)} = \arg\min_{\mathbf{f}^\top} \; \Psi(\mathbf{f} \mid F^{(t)}, i) \quad \textbf{s.t. } \mathbf{f} \geq 0 \tag{13}$$

where the objective function Ψ is defined as follows:

$$\begin{aligned} \Psi(\mathbf{f} \,|F,i) &= L_{ii}\|\mathbf{f}\|^2 + 2\hat{L}_{i*}\hat{F}\mathbf{f} + \alpha\mathbf{z}^\top R\mathbf{z} + \beta\mathbf{f}^\top T\mathbf{f} \\ &= \mathbf{f}^\top\hat{Q}\mathbf{f} + \hat{\mathbf{c}}^\top\mathbf{f} + \hat{\mathbf{h}} \end{aligned}$$

where $\hat{L}_{i*} \in \mathbb{R}^{1\times(n-1)}$ is the i-th row of Laplacian matrix L_{i*} by removing the i-th element L_{ii}, $\hat{F} \in R^{(n-1)\times m}$ is a sub-matrix of F by removing its i-th row F_{i*}, $\mathbf{z} = \mathbf{f} - Y_{i*}^\top$, $R = diag(S_{i*})$, $T = \mathbf{1}\cdot\mathbf{1}^\top$, $\hat{Q} = L_{ii}I_M + \alpha R + \beta T$, $\hat{\mathbf{c}} = 2(\hat{L}_{i*}\hat{F} - \alpha Y_{i*}R)^\top$ and $\hat{\mathbf{h}} = \alpha Y_{i*}RY_{i*}^\top$.

The Eq.13 is also a smooth QP problem, but much smaller than the original Eq.7. Similarly, it could be solved efficiently by Algorithm 1. The pseudo-code of the coordinate descent algorithm is summarized in Algorithm 2.

```
Algorithm 2: Coordinate Descent Algorithm for ULR
Input: $X \in \mathbb{R}^{n \times d}$, $Y \in [0,1]^{n \times m}$
Output: $F^{\star} \in \mathbb{R}^{n \times m}$
begin
    $t = 0$ and $F^{(t)} = Y$;
    repeat
        for $i = 1$ to $n$ do
            Case SRF: Achieve $F_{i*}^{(t+1)}$ with Eq. (13);
            Case CCF: Achieve $F_{i*}^{(t+1)}$ with Eq. (21);
        $t = t + 1$;
    until CONVERGENCE;
```

4.3.2 Algorithm for Convex-Constraint Formulation

For the convex-constraint formulation, by doing vectorization, we can reformulate Eq. (6) into the following:

$$\min_{\mathbf{x} \geq 0} \mathbf{x}^{\top} Q^{\dagger} \mathbf{x} + \mathbf{c}^{\top} \mathbf{x} \quad \text{s.t.} \sum_{k=0}^{m-1} x_{k \cdot n+i} \leq \varepsilon, i = 1, \ldots, n. \tag{14}$$

where $Q^{\dagger} = U + \alpha R$, $\varepsilon \geq 1$, and all the other symbols are the same as Eq. (7). We also apply the multi-step gradient scheme to solve Eq. (14), however the constraint for the sub-problem is slightly different from Eq. (11), which is defined:

$$\min_{\mathbf{x} \geq 0} \frac{t^{\dagger}}{2} \|\mathbf{x} - \mathbf{v}\|^2 \quad \text{s.t.} \sum_{k=0}^{m-1} x_{k \cdot n+i} \leq \varepsilon, i = 1 \ldots, n \tag{15}$$

where $\mathbf{v} = \mathbf{z}^{(k)} - \frac{1}{t^{\dagger}} \mathbf{g}^{\dagger}$, $\mathbf{g}^{\dagger} = 2Q^{\dagger}\mathbf{z}^{(k)} + \mathbf{c}$.

We can split $\mathbf{x}$ into a series of sub-vectors $\bar{\mathbf{x}}^i = [x_i, \ldots, x_{(m-1)*n+i}]^{\top}$, and similarly we can split vector $\mathbf{v}$. Thus, Eq. (15) could be reformulated as:

$$\min_{\bar{\mathbf{x}}^0, \bar{\mathbf{x}}^1, \ldots, \bar{\mathbf{x}}^n} \frac{t^{\dagger}}{2} \sum_{i=1}^{n} \|\bar{\mathbf{x}}^i - \bar{\mathbf{v}}^i\|^2 \quad \text{s.t.} \|\bar{\mathbf{x}}^i\|_1 \leq \varepsilon, \ \bar{\mathbf{x}}^i \geq 0 \tag{16}$$

The above optimization can be decoupled for each sub-vector $\bar{\mathbf{x}}^i$ and solved separately in linear time by following the Euclidean projection algorithm proposed in [58]. Specifically, we can obtain the optimal solution $\bar{\mathbf{x}}^{i\star}$ for $\bar{\mathbf{x}}^i$ with the following problem:

$$\bar{\mathbf{x}}^{i\star} = \arg\min_{\bar{\mathbf{x}}^i} \|\bar{\mathbf{x}}^i - \bar{\mathbf{v}}^i\|^2 \quad \text{s.t.} \quad \|\bar{\mathbf{x}}^i\| \leq \varepsilon; \bar{\mathbf{x}}^i \geq 0. \tag{17}$$

where $\bar{\mathbf{x}}^{i\star}$ has a linear relationship with the optimal Lagrangian variable $\lambda^{\star}$, which is introduced by the inequlaity constrain $\|\bar{\mathbf{x}}_i\| \leq \varepsilon$:

$$\bar{x}_j^{i\star} = sign(\bar{v}_j^i) \times \max(|\bar{v}_j^i| - \lambda^{\star}, 0), j = 1, 2, ...m. \tag{18}$$

where $sign(\cdot)$ is the previously defined sign function. Suppose $S = \{j | \bar{v}_j^i \geq 0\}$, the optimal $\lambda^{\star}$ could be obtained as follows:

$$\lambda^{\star} = \begin{cases} 0 & \sum_{k \in S} |\bar{v}_k^i| \leq \varepsilon, \\ \bar{\lambda} & \sum_{k \in S} |\bar{v}_k^i| > \varepsilon. \end{cases} \tag{19}$$

where $\bar{\lambda}$ is the unique root of function $f(\lambda)$:

$$f(\lambda) = \sum_{k \in S} \max(|\bar{v}_k^i| - \lambda, 0) - \varepsilon. \tag{20}$$

$f(\lambda)$ is a continuous and monotonically decreasing function in $(-\infty, \infty)$. The root $\bar{\lambda}$ could be achieved with a bisection search in linear time. An improved searching scheme is also proposed in [58] by using the characteristic of function $f(\lambda)$, which is out of the scope of this paper.

Similar to the soft-regularization formulation, we can also adopt the coordinate descent scheme to further improve the scalability. In particular, we define a new update function $\Psi^{\dagger}$ similar to the aforementioned formula in Eq. (13):

$$F_{i*}^{(t+1)} = \arg\min_{\mathbf{f}^{\top}} \Psi^{\dagger}(\mathbf{f} \mid F^{(t)}, i) \ \text{s.t.} \ \|\mathbf{f}\|_1 \leq \varepsilon, \mathbf{f} \geq 0 \tag{21}$$

where $\Psi^{\dagger}(\mathbf{f} \mid F, i) = \mathbf{f}^{\top} \hat{Q}^{\dagger} \mathbf{f} + \hat{\mathbf{c}}^{\top} \mathbf{f}$ and all symbols are the same as Eq. (13) except $\hat{Q}^{\dagger} = L_{ii} I_M + \alpha R$. Eq. (21) is a special case of the optimization problem in Eq. (14), and can be solved efficiently by the same algorithm. Finally, the pseudo codes of the algorithm for the convex-constraint formulation are similar to the previous, as shown in Algorithm 1 and Algorithm 2, respectively.

4.4 Clustering-based Approximation (CBA)

The number of variables in the previous problem is $n * m$, where n is the number of facial images in the retrieval database and m is the number of distinct names (classes). For a small problem, we can solve it efficiently by the proposed MGA-based algorithms (SRF-MGA or CCF-MGA). For a large problem, we can adopt the proposed CDA-based algorithms (SRF-CDA or CCF-CDA), where the number of variables in each sub-problem is n. However, when n is extremely large, the CDA-based algorithms still can be computationally intensive. One straightforward solution for acceleration is to adopt parallel computation, which can be easily exploited by the proposed SRF-CDA or CCF-CDA algorithms since each of the involved sub-optimization tasks can be solved independently. However, the speedup of the parallel computation approach quite depends on the hardware capability. To further enhance the scalability and efficiency in algorithms, in this paper, we propose a Clustering-based Approximation (CBA) solution to speed up the solutions for large-scale problems.

In particular, the clustering strategy could be applied in two different levels: (i) one is on "image-level", which can be used to directly separate all the n facial images into a set of clusters, and (ii) the other is on "name-level", which can be used to firstly separate the m names into a set of clusters, then to further split the retrieval database into different subsets according to the name-label clusters. Typically, the number of facial images n is much larger than the number of names m, which means that the clustering on "image-level" would be much more time-consuming than that on "name-level". Thus, in our approach, we adopt the "name-level" clustering scheme for the sake of scalability and efficiency. After the clustering step, we solve the proposed ULR problem in each subset, and then merge all the learning results into the final enhanced label matrix F.

According to the name labels $\{n_1, n_2, \dots, n_m\}$, we could divide all the facial images $X \in \mathbb{R}^{n\times d}$ into m classes: $X = [X_1, X_2, \dots, X_m]$. We denote by $C \in \mathbb{R}^{m\times m}$ the class similarity matrix for all the m classes (names). Consider the variety of facial images and the noisy nature of web images, traditional hierarchical clustering algorithms (such as "Single-Link", "Complete-Link" and "Average-Link") are not suitable to our problem. In our framework, following the terminology of shared nearest neighbors, we proposed a *co-occurrence likelihood* in Eq. (22) to compute the similarity value C_{ij}, which measures the likelihood that instances from the two classes X_i,X_j are co-occurred together in the retrieval results by some particular web search engine (e.g., Google):

$$C_{ij} = \sum_{\forall \mathbf{x}_i \in X_i} \sum_{x_p \in \mathcal{N}_K(\mathbf{x}_i)} \mathbb{I}_{(\mathbf{x}_p \in X_j)} \tag{22}$$

where $\mathcal{N}_K(\mathbf{x}_i)$ is the set of top K nearest facial images w.r.t. $\mathbf{x}_i$ in the whole retrieval database (we use the nearest facial set $\mathcal{N}_K(\mathbf{x}_j)$ with $K = 50$ in our experiments), $\mathbb{I}_{(\mathbf{x}_p \in X_j)}$ is an indicator function which outputs 1 if $x_p \in X_j$ and 0 otherwise. According to this definition, a large value of C_{ij} means that the instances in class X_i are more likely to be similar to the instances in class X_j. In other words, the instances in X_i and X_j should be put together for joint class label refinement in our proposed label enhancement step. In order to normalize the elements in the matrix C, we divide each column $C_{\star j}$ by its maximum value except C_{jj}:

$$C_{ij} = \begin{cases} \frac{C_{ij}}{\max_{k\neq i} C_{kj}} & \text{if } j \neq i, \\ v_{\mathtt{max}} & \text{if } j = i. \end{cases} \tag{23}$$

where $v_{\mathtt{max}}$ is a constant value and set as $v_{\mathtt{max}} = 1$ in our experiment. Fig. 2 shows an example to demonstrate the calculation of matrix C among three classes X_1, X_2 and X_3 with $K = 1$. After the normalization, the co-occurrence likelihood vectors for the three classes are [110], [110] and [001], which are consistent to our observation that instances from class C_1 and C_2 are more likely to be mixed together.

For the proposed solution, there is an important practical assumption for the clustering step, i.e., the sizes of different clusters should be similar, which aims to avoid the undesired case where one cluster significantly dominates the others. In our CBA framework, we propose two kinds of solutions: one is the Bisecting K-means clustering based algorithm referred to as "BCBA" for short, and the other is the Divisive Clustering based algorithm referred to as "DCBA" for short.

In the BCBA scheme, the i-th row $C_{i\star}$ is used as the feature vector for class X_i. In each step, the largest cluster is bisected for $I_{\mathtt{loop}}$ times and the clustering result with the lowest sum-of-square-error (SSE) value is used to update the clustering lists. In our framework, we set $I_{\mathtt{loop}}$ to 10. The details of the BCBA scheme are illustrated in Algorithm 3, where q_c is the cluster number. In the DCBA scheme, the symmetrical matrix $\hat{C} = \frac{C+C'}{2}$ is used for building a minimum spanning tree (MST). Instead of performing the complete hierarchical clustering, in our framework, we directly separate the classes into the q_c clusters. In order to balance the cluster sizes, the bisection scheme is also employed. Specifically, in each iteration step, we partition the largest cluster into two parts by cutting its largest MST edge to ensure the size of the smaller cluster in the cutting result is larger than a predefined threshold value $T_{\mathtt{threshold}}$. We set $T_{\mathtt{threshold}} = \frac{m}{2*q_c}$ in our framework. The details of the DCBA scheme are shown in Algorithm 4.

We denote by $L_{\mathtt{list}} = \{\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_{q_c}\}$ the clustering result, where $\mathcal{M}_{i=1,2,\dots,q_c} \subseteq \Omega$. Using the clustering result, we first split the whole retrieval database X into q_c subsets $\{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_{q_c}\}$, where $\mathcal{S}_i = \bigcup_{n_j \in \mathcal{M}_i} X_j$. Then the proposed ULR problem is conquered on each subset individually. For each subset $\mathcal{S}_i$, the number of classes is around $\frac{M}{q_c}$ on average and the number of facial images is around $\frac{N}{q_c}$ on average, which means that the number of variables to be optimized by ULR on each subset $\mathcal{S}_i$ has been reduced to $\frac{N\times M}{q_c^2}$, which is much smaller than the original number of variables on the entire database. As a result, each small sub-problem could be solved efficiently. Besides, as the sub-problems on different subsets are independent, a parallel computation framework could also be adopted for further acceleration.

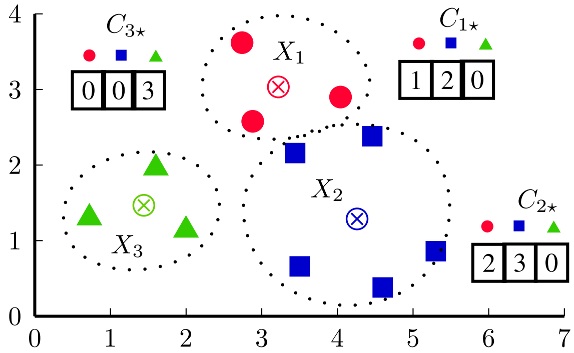


Fig. 2. Illustration of computing class similarity matrix between three classes X_1, X_2 and X_3. The symbol $\otimes$ denotes the class center. $C_{1\star}$, $C_{2\star}$ and $C_{3\star}$ are the similarity vectors of the three classes, which are computed according to Eq. (22) with $K = 1$. For example, the second value of vector $C_{1\star}$, i.e., C_{12}, refers to the total number of examples in class X_2 belonging to the top $K = 1$ nearest neighbors of examples from class X_1.

Algorithm 3: Bisecting K-means Clustering-based Approximation (BCBA)

```
Input: $C \in \mathbb{R}^{m\times m}$, $q_c \in \mathbb{N}$, $I_{\mathtt{loop}} \in \mathbb{N}$.
Output: Clustering result list $L_{\mathtt{list}}$
Add $\mathcal{M}_0$ to $L_{\mathtt{list}}$; /*$\mathcal{M}_0$ contains all the points*/
repeat
    Remove the largest cluster $\mathcal{M}_l$ from $L_{\mathtt{list}}$;
    for $i = 1$ to $t$ do
        Bisect $\mathcal{M}_l$ to $\mathcal{M}_1^{(i)}$ and $\mathcal{M}_2^{(i)}$;
        Compute Sum of Squared Error(SSE$_i$);
    Select the result with the lowest SSE$_i$ value;
    Add $\mathcal{M}_1^{(i)}$, $\mathcal{M}_2^{(i)}$ to $L_{\mathtt{list}}$;
until $|L_{\mathtt{list}}| = q_c$;
```

Algorithm 4: Divisive Clustering-based Approximation(DCBA)

```
Input: $\hat{C} \in \mathbb{R}^{m\times m}$, $q_c \in \mathbb{N}$, $T_{\texttt{threshold}} \in \mathbb{N}$.
Output: Clustering result list $L_{\texttt{list}}$
Build MST $T_{\texttt{tree}}$ with $\hat{C}$ and add all edges to $E_{\texttt{edge}}$;
Add $\mathcal{M}_0$ to $L_{\texttt{list}}$; /*$\mathcal{M}_0$ contains all the points*/
while $|L_{\texttt{list}}| < q_c$ do
    Remove the largest cluster $\mathcal{M}_l$ from $L_{\texttt{list}}$;
    Extract sub-MST $T^{\dagger}_{\texttt{tree}}$ and edge set $E^{\dagger}_{\texttt{edge}}$ of $\mathcal{M}_l$;
    repeat
        if $|E^{\dagger}_{\texttt{edge}}| == 0$ then
            Rebuild edge list $E^{\dagger}_{\texttt{edge}}$ with $T^{\dagger}_{\texttt{tree}}$;
            $T_{\texttt{threshold}} = T_{\texttt{threshold}} - 1$;
        Remove the largest edge $e^{\S}$ from $E^{\dagger}_{\texttt{edge}}$;
        Cutting $e^{\S}$ in $T^{\dagger}_{\texttt{tree}}$ to split $\mathcal{M}_l$ into $\mathcal{M}_1$, $\mathcal{M}_2$
    until $\min(|M_1|, |M_2|) \geq T_{\texttt{threshold}}$;
    Add $\mathcal{M}_1$, $\mathcal{M}_2$ to $L_{\texttt{list}}$;
```

5 Experiments

5.1 Experiment Testbed

In our experiments, we collected a human name list consisting of popular actor and actress names from the **IMDb** website: http://www.imdb.com. In particular, we collected these names with the billboard: "Most Popular People Born In **yyyy**" of **IMDb**, where **yyyy** is the born year. e.g. the webpage [2] presents all the actor and actresses who were born in 1975 in the popularity order. Our name list covers the actors and actresses who were born between 1950 and 1990. To enlarge the retrieval database, we extended the name number in [13] from 400 to 1000. We submitted each name from the list as a query to search for the related web images by Google image search engine. The top 200 retrieved web images are crawled automatically. After that we used the OpenCV toolbox to detect the faces and adopt the DLK algorithm [54] to align facial images into the same well-defined position. The no-face-detected web images were ignored. As a result, we collected over $100,000$ facial images in our database. We refer to this database as the "retrieval database", which will be used for facial image retrieval during the auto face annotation process. In order to evaluate varied number of persons in database, we divided our database into two scales: one contains 400 persons and about $40,000$ and the other contains 1000 persons and about $100,000$ images. We denote them by "DB0400" and "DB1000" respectively.

For the "test dataset", we used the same testset in [13]. Specifically, we randomly chose 80 names from our name list. We submitted each selected name as a query to Google and crawled about 100 images from the top 200-th to 400-th search results. Note that we did not consider the top 200 retrieved images since they had already appeared in the retrieval dataset. This aims to examine the generalization performance of our technique for unseen facial images. Since these facial images are often noisy, to obtain ground truth labels for the test dataset, we request our staff to manually examine the facial images and remove the irrelevant facial images for each name. As a result, the test database consists of about 1000 facial images with over 10 faces per person on average. The data sets and code of this work can be downloaded from http://www.cais.ntu.edu.sg/~chhoi/ULR/.

5.2 Comparison Schemes and Setup

In our experiments, we implemented all the algorithms described previously for solving the proposed ULR task. We finally adopted the soft-regularization formulation of the proposed ULR technique in our evaluation since it is empirically faster than the convex-constraint formulation according to our implementations. To better examine the efficacy of our technique, we also implemented some baseline annotation method and existing algorithms for comparisons. Specifically, the compared methods in our experiments include the following:

- "ORI": a baseline method that simply adopts the original label information for the search-based annotation scheme, denoted as "ORI" for short.
- "CL": a consistency learning algorithm [48] proposed to enhance the weakly labeled facial image database, denoted as "CL" for short.
- "MKM": a modified K-means clustering algorithm [5] proposed to cluster web facial images associated with the extracted names from the surrounding captions, denoted as "MKM" for short. We note that the original MKM algorithm was proposed to address a similar noisy label enhancement problem, but slightly different from our setting in that the number of raw noisy labels of each facial image in their problem setting can be more than 1, which is however exactly equal to 1 in our problem setting.
- "LPSN": a label propagation through sparse neighborhood algorithm [59] proposed to propagate label information among the neighborhoods achieved by sparse coding, denoted as "LPSN" for short.
- "ULR$_{\beta=0}$": the proposed ULR algorithm (Soft-Regularization Formulation in Eq. 5) without the sparsity regularizer $E_e(F)$.
- "ULR": the proposed unsupervised label refinement method, denoted as "ULR" for short.

To evaluate their annotation performances, we adopted the *hit rate* at top t annotated results as the performance metric, which measures the likelihood of having the true label among the top t annotated names. For each query facial image, we retrieved a set of top K similar facial images from the database set, and return a set of top T names for annotation by performing a majority voting on the labels associated with the set of top K images.

Further, we discuss parameter settings. For the ULR implementation, we constructed the sparse graph W by setting the number of nearest neighbors to 5 for all cases. In addition, for the two key regularization parameters α and β in the proposed ULR algorithm, we set their values via cross validation. In

[2] http://www.imdb.com/search/name?birth_year=1975

particular, we randomly divided the test dataset into two equally-sized parts, in which one part was used as validation to find the optimal parameters by grid search, and the other part was used for testing the performance. This procedure was repeated 10 times, and their average performances were reported in our experiments.

5.3 Evaluation of Facial Feature Representation

In this experiment, we evaluate the face annotation performance of five types of facial features for the baseline ORI algorithm. TABLE 1 shows the annotation performance. All of these features are extracted from the aligned facial images by the DLK algorithm [54], as shown in Fig. 3.

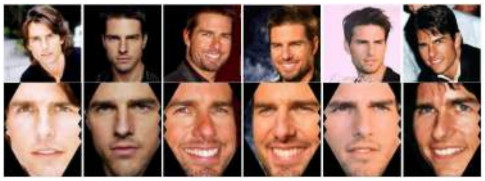


Fig. 3. The examples of web facial images and the corresponding alignment results with DLK algorithm.

The "Gist", "Edge", "Color", and "Gabor" features are generated by the FElib toolbox [3]. For the "LBP" feature, the aligned facial image is divided into 7 × 7 windows [60] resulting a 2891-dimension feature. From our experimental results, it is clear to observe that GIST is much or at least slightly better than the other common features. The "LBP" feature is highly closed to the "Gist" feature, however its feature dimension is much higher. If we projected the original "LBP" feature into a low-dimensional space that is the same with the "GIST" feature, denoted as "LBP-PCA512", the performance nevertheless decreases significantly. In the following experiments, for a fair comparison, we adopted the same GIST features to represent the facial images.

TABLE 1
The performance of the baseline ORI algorithm with different facial feature representations.

Gist	Edge	Color	Gabor	LBP	LBP-PCA512
0.548 ±0.013	0.154 ±0.012	0.238 ±0.013	0.345 ±0.011	0.535 ±0.010	0.510 ±0.012

5.4 Evaluation of Auto Face Annotation

In this experiment, we aim to evaluate the auto face annotation performance based on the search-based face annotation scheme. We firstly evaluated the proposed ULR algorithm from different aspects on database "DB0400" with top 100 retrieval facial images per person, and then verified its performance on the large-scale database "DB1000". TABLE 2 and Figure. 4 show the average annotation performance (hit rates), in which both mean and standard deviation were reported. Several observations can be drawn from the results.

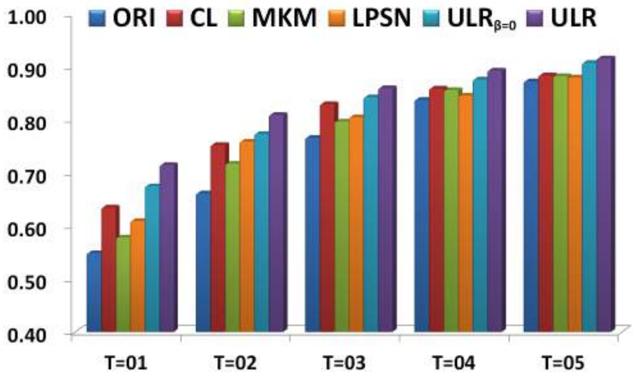


Fig. 4. Evaluation of auto face annotation performance in terms of *hit rates* at top T annotated names.

First of all, it is clear that ULR which employs unsupervised learning to refine labels consistently performs better than the ORI baseline using the original weak label, the existing CL algorithm, MKM algorithm, and the LPSN algorithm. The promising result validates that the proposed ULR algorithm can effectively exploit the underlying data distribution of all data examples to refine the label matrix and improve the performance of the search-based face annotation approach. Second, we note that the ULR algorithm outperforms its special case "URL$_{\beta=0}$" without the sparsity regularizer in the SRF formulation, which validates the importance of the sparsity regularizer. Finally, when T is small, the hit rate gap, i.e., the hit rate difference between ORI and ULR is more significant, and the annotation performance increases slowly when T is large. In practice, we usually focused on the small T value since users typically would not be interested in a long list of annotated names.

TABLE 2
Evaluation of auto face annotation performance in terms of *hit rates* at top T annotated names.

	T=01	T=02	T=03	T=04	T=05
ORI	0.548 ± 0.013	0.661 ± 0.011	0.766 ± 0.009	0.837 ± 0.010	0.872 ± 0.010
CL	0.634 ± 0.012	0.752 ± 0.010	0.829 ± 0.010	0.858 ± 0.010	0.883 ± 0.009
MKM	0.578 ± 0.011	0.717 ± 0.012	0.797 ± 0.012	0.856 ± 0.008	0.882 ± 0.010
LPSN	0.609 ± 0.015	0.758 ± 0.016	0.805 ± 0.015	0.845 ± 0.016	0.879 ± 0.014
ULR$_{\beta=0}$	0.675 ± 0.015	0.773 ± 0.017	0.842 ± 0.016	0.875 ± 0.015	0.907 ± 0.015
ULR	0.715 ± 0.008	0.809 ± 0.005	0.859 ± 0.009	0.892 ± 0.007	0.916 ± 0.010

5.5 Evaluation on Varied Top K Retrieved Images and Top T Annotated Names

This experiment aims to examine the relationship between the annotation performance of varied values of K and T respectively for top K retrieved images and top T annotated names. To ease our discussion, we only show the results of the ULR algorithm. The face annotation performance of varied K and T values are illustrated in Fig. 5.

[3] http://goo.gl/BzPPx

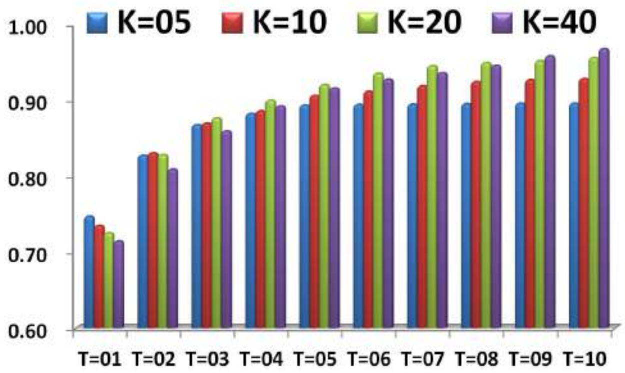
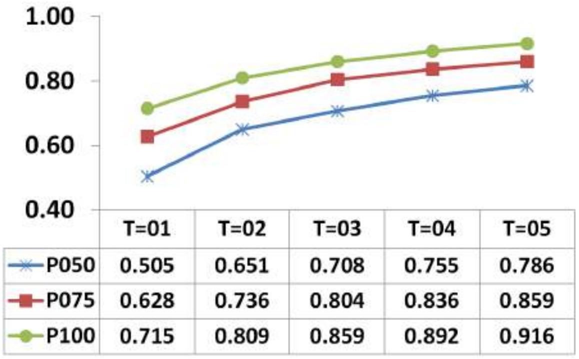


Fig. 5. Annotation performance w.r.t. varied K and T values.



	T=01	T=02	T=03	T=04	T=05
P050	0.505	0.651	0.708	0.755	0.786
P075	0.628	0.736	0.804	0.836	0.859
P100	0.715	0.809	0.859	0.892	0.916

Fig. 6. The annotation performance on three different databases, which have different numbers of images per person. Specifically, P050, P075, and P100 denote the databases have the top 50, 75, and 100 retrieval images per person, respectively.

Some observations can be drawn from the experimental results. First of all, when fixing K, we found that increasing T value generally leads to better hit rate results. This is not surprising since generating more annotation results certainly gets a better chance to hit the relevant name. Second, when fixing T, we found that the impact of the K value to the annotation performance fairly depends on the specific T value. In particular, when T is small (e.g. $T = 1$), increasing the K value leads to the decline of the annotation performance; but when T is large (e.g. $T > 5$), increasing the K value often boosts the performance of top T annotation results. Such results can be explained as follows. When T is very small, e.g. $T = 1$, we prefer a small K value such that only the most relevant images will be retrieved, which thus could lead to more precise results at top-1 annotated results. However, when T is very large, we prefer a relatively large K value since it can potentially retrieve more relevant images and thus can improve the hit rate at top T annotated results.

5.6 Evaluation on Varied Numbers of Images per Person in Database

This experiment aims to further examine the relationship between the annotation performance and the number of facial images per person in building the facial image database. Unlike the previous experiment with top 100 retrieval facial images per person in the database, we created three variables of varied-size databases, which consist of top 50, 75, and 100 retrieval facial images per person, respectively. We denote these three databases as P050, P075, and P100, respectively.

Fig. 6 shows the experiment results of average annotation performance. It is clear that the larger the number of facial images per person collected in our database, the better the average annotation performance can be achieved. This observation is trivial since more potential images are included into the retrieval database, which is beneficial to the annotation task. We also noticed that enlarging the number of facial images per person in general leads to the increases of computational costs, including time and space costs for indexing and retrieval as well as the ULR learning costs.

5.7 Evaluation on a Larger Database: DB1000

This experiment aims to verify the annotation performance of the proposed SBFA framework over a larger retrieval database: "DB1000". As the test database is unchanged, the extra facial images in the retrieval database are definitely harmful to the nearest facial retrieval result for each query image. A similar result could also been observed in [47], where the mean average precision became smaller for a larger retrieval database. As a result, the final annotation performance of DB1000 would be worse than the one over DB0400. More details of the experiment are presented in Fig. 7.

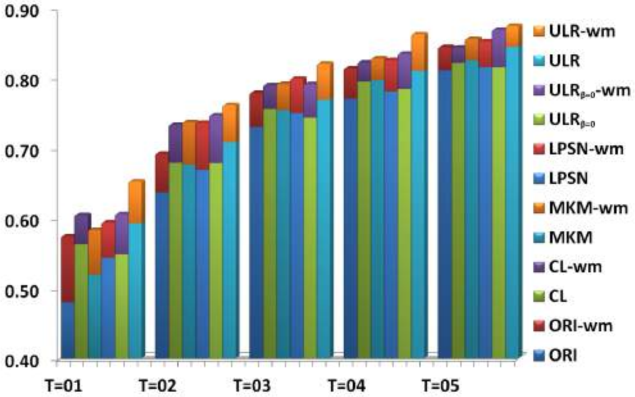


Fig. 7. Face annotation performance on Database: DB1000. The algorithms that end up with "-wm" denote the improved performances achieved by weighted majority voting method in the name annotation step.

Some observations can be drawn from the experimental results. First of all, the proposed ULR algorithm also could efficiently enhance the initial noisy label and achieve the best performance over the other algorithms. Secondly, all the algorithms perform slightly worse on the larger retrieval database. In detail, the ULR annotation performance on DB1000 is about 0.83% of the one on DB0400.

In order to further improve the system performance, we adopt a simple weighted majority voting scheme in the third step of Fig. 1. Specifically, we assign a weighting value to each facial image in the short list of similar faces according to its ranking position : $w(n;q,\sigma) = e^{-\frac{(n-1)^q}{\sigma}}$, where n is the ranking position and $q > 0$, $\sigma > 0$ are two positive parameters. Obviously the larger n is, the smaller the weighting $w(n;q,\sigma)$ is, which means less contribution is introduced for the label annotation. The improved performance is also presented in Fig. 7. The main observation is the annotation performance can be significantly improved, for example, the performance of ULR is boosted to 65.2% from 59.3%. This experiment also illustrates that the performance of our current SBFA system can be further improved by adopting other more sophisticated techniques in different stages of the proposed solution, which is out of the scope of our focus in this paper.

5.8 Evaluation of Optimization Efficiency

This section aims to conduct extensive evaluations on the running time cost by the four different algorithms. We refer the four algorithms with the following abbreviations:

- SRF-MGA: Soft-Regularization Formulation solved by the Multi-step Gradient Algorithm.
- SRF-CDA: Soft-Regularization Formulation solved by the Coordinate Decent Algorithm.
- CCF-MGA: Convex-Constraint Formulation solved by the Multi-step Gradient Algorithm.
- CCF-CDA: Convex-Constraint Formulation solved by the Coordinate Decent Algorithm.

We first compared two algorithms: SRF-MGA and CCF-MGA, which adopt the same gradient-based optimization scheme for two different formulations, as shown in Algorithm 1. We used an artificial dataset with varied numbers of classes $m =$ 20, 40, 60, 80, 100 where each class corresponds to a unique Gaussian distribution. We set the number of examples generated from each class as $P = 100$, and the total number of examples $n =$2K, 4K, 6K, 8K, 10K. The goal of our ULR optimization task is to optimize the refined label matrix $F \in \mathbb{R}^{n \times m}$, which has the total number of unknown variables V would be 40K, 160K, 360K, 640K, 1M, respectively for each of the above cases. For the iteration number, we set it to 50 for both algorithms.

We randomly generated the artificial dataset and run the algorithms over the dataset. This procedure was repeated five times. The first two columns of TABLE 3 show the average running time cost obtained by both SRF-MGA and CCF-MGA algorithms, respectively. We observed that the time cost growth rate of SRF-MGA is always slower than that of CCF-MGA, which indicates that SRF-MGA runs always more efficiently than CCF-MGA. To further compare the difference of their growth rates, we try to fit the running time costs T with respect to the number of variables V by a function $T = a \times V^b$, where $a, b \in \mathbb{R}$ are two parameters. By fitting the functions, we obtained $a = 9.04E-7$ and $b = 1.45$ for SRF-MGA, and $a = 3.70E-8$ and $b = 1.74$ for CCF-MGA.

Next, we compare running time cost of RF-CDA and CCF-CDA by adopting the similar settings as the previous experiment. For the iteration number, we set the outer-loop iteration number for CDA to 30 and fix the inner iteration number with respect to their subproblems to 30. The average running time cost is illustrated in the last two columns of TABLE 3

TABLE 3
The average running time (seconds) of the four proposed algorithms.

V	SRF-MGA	CCF-MGA	SRF-CDA	CCF-CDA
40000	4.80	6.76	76.63	136.50
160000	29.04	49.05	197.92	330.30
360000	95.43	178.45	399.11	607.20
640000	228.44	494.29	670.23	950.40
1000000	428.46	1076.04	1022.70	1457.40

First, we found that the SRF-based algorithm SRF-CDA spent less time cost than the CCF-based algorithm CCF-CDA. Second, the running time cost grows almost linearly with respect to the number of variables for both CDA based algorithms. More specifically, by fitting the time cost function $T = a \times V^b$ with respect to the number of variables V, we have $a = 3.93E-3$ and $b = 0.90$ for SRF-CDA, and $a = 1.50E-2$ and $b = 0.83$ for CCF-CDA, which showed that the time cost growth rates of both algorithms are empirically sublinear. This encouraging result indicates that both CDA based algorithms are efficient and scalable for large-scale dataset.

5.9 Evaluation of Clustering-based Approximation

In this experiment, we aim to evaluate the acceleration performance of the two proposed Clustering-based Approximation (CBA) schemes (BCBA and DCBA) on the large database DB1000. A good approximation is expected to achieve a high reduction in running time with a small loss in annotation performance. Thus, this experiment evaluates both running time and annotation performance.

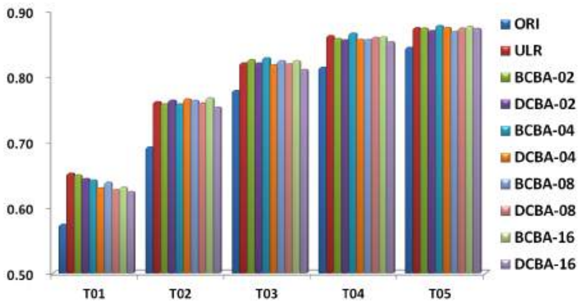


Fig. 8. Evaluation of annotation performance archived by Clustering-based Approximation (CBA) with different methods (BCBA, DCBA) and different group numbers K; $K = 1$ is the intact scheme without acceleration.

The running time of CBA scheme mainly consists of three parts : (i) the time of constructing the similarity matrix C; (ii) the time of clustering; (iii) the total time of running ULR algorithm in each subset. The running time costs of different clustering algorithms with different cluster numbers ($q_c = 02, 04, 08, 16$) are illustrated in TABLE 4. As a comparison, the running time of directly adopting ULR algorithm on the

TABLE 4
Evaluation of running time used by Clustering-based Approximation (CBA)

	ULR	Build	$q_c = 02$		$q_c = 04$		$q_c = 08$		$q_c = 16$	
	$q_c = 01$	C	BCBA	DCBA	BCBA	DCBA	BCBA	DCBA	BCBA	DCBA
T_{C}	—	60.70 ± 0.70	1.82 ± 0.17	0.85 ± 0.01	3.80 ± 0.18	0.96 ± 0.01	5.43 ± 0.21	1.19 ± 0.04	7.06 ± 0.21	1.36 ± 0.01
T_{U}	26628.94 ± 768.06	—	7068.59 ± 259.19		3357.14 ± 62.25		2933.77 ± 75.41		2654.69 ± 67.44	
T_{T}	26628.94	—	7131.12	7130.14	3421.64	3418.81	2999.90	2995.67	2722.46	2716.76
%	—	—	26.78%	26.78%	12.85%	12.84%	11.27%	11.25%	10.22%	10.20%

whole retrieval database is also presented in the second column of TABLE 4, denoted as "URL ($q_c = 01$)". Some observations can be drawn from these results.

First of all, the proposed CBA scheme could significantly decrease the running time for the label refinement task. For example, for BCBA and DCBA schemes with $q_c = 02$, the total running time could reduced from about $26,629$ seconds to $7,131 (27\%)$ seconds and $7,130 (27\%)$ seconds respectively.Secondly, increasing the value of cluster number q_c generally leads to less running time, however, the reduction becomes marginal where q_c is larger than some threshold(e.g. $q_c = 08$). Thirdly, the running time of the division clustering algorithm is a bit smaller than the one of bisecting K-mean algorithm. The reasons leading to this phenomenon are twofold: one is there is no need for multi loops in each bisection step of DCBA, another is the similarity matrix $\hat{C}$ is directly used for MST building without extra computation.

For the annotation performance, the weighted majority annotation result of the two CBA schemes (BCBA and DCBA) with different cluster number q_c are presented in TABLE 5 and Fig. 8. Two observations can be drawn from the results.

Firstly, although the approximation algorithms (BCBA, DCBA) slightly degrade the final annotation performance, their performances are still much better than the other compared algorithms for small T value. Considering the reduction in running time, the proposed clustering-based approximation scheme is a good approximation for the ULR algorithm, which could significantly improve the scalability of search-based face annotation framework. Secondly, the performance difference between BCBA and DCBA are statistically marginal, but the average performance of BCBA is a bit better than DCBA.

5.10 Label Refinement on Artificial Dataset

In this experiment, we aim to evaluate the label refinement performance of different algorithms. We built an artificial dataset that consists of 9 classes (persons) in 2-dimensional space with 20 samples for each class. To introduce noise into the label matrix, we randomly mislabeled half of the whole dataset. All the data points are illustrated in Fig. 9(a), and the original noisy label matrix is shown as the leftmost one in Fig. 9(b). Given the dataset and the noisy label matrix, we computed the enhanced label matrixes using the four algorithms mentioned in Section 5.2(see Fig. 9(b)).

Several observations can be drawn from the above results: first, the MKL and CL algorithms work well for the classes with less noise(e.g. Person 1 and Person 9), but they fail for the classes where more samples are mislabeled and widely distributed (e.g. Person 4 and Person 5). Second, by adopting the graph information, both LPSN and ULR could handle all the classes better. Obviously, by finding the maximum value in each label vector, we can recover the ideal label matrix from the refined label matrix F_{ULR}. Third, for the proposed ULR algorithm, we also consider the distortion with the original label matrix ($E_p(F, Y)$ in Eq.5) and the sparsity of each label vector ($E_e(F)$ in Eq.5). As a result, ULR can achieve more stable and sparse refined label matrix that is more suitable for our face annotation problem.

6 Limitations

Our work is limited in several aspects. First, we assume each name corresponds to a unique single person. Duplicate name can be a practical issue in real-life scenarios. We can extend our method to address this practical problem. For example, we can learn the similarity between two different names so as to determine how likely the two different names belong to the same person. Second, we assume the top retrieved web facial images are related to a query human name. This is clearly true for celebrities. However, when the query facial image is not a well-known person, there may not exist many relevant facial images on the WWW. This is a common limitation of all existing data-driven annotation techniques. This might be partially solved by exploiting social contextual information.

7 Conclusions

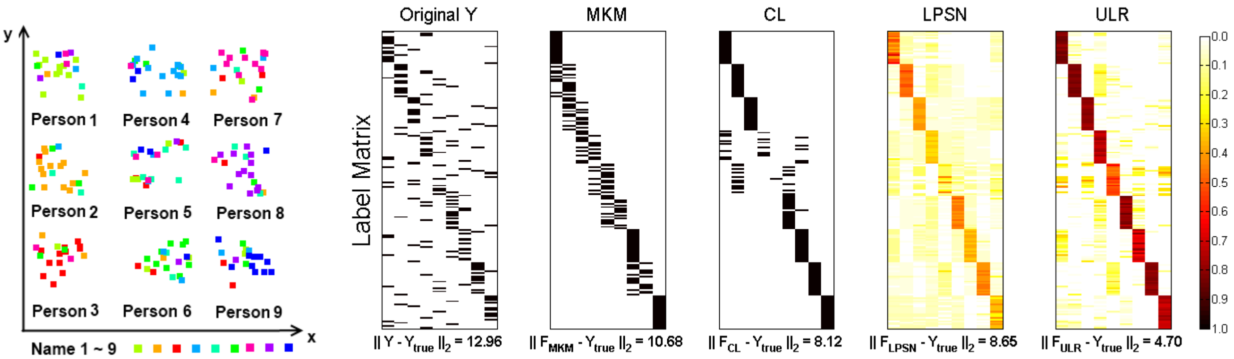
This paper investigated a promising search-based face annotation framework, in which we focused on tackling the critical problem of enhancing the label quality and proposed an Unsupervised Label Refinement (ULR) algorithm. We also proposed a Clustering-based Approximation (CBA) solution, which successfully accelerated the optimization task without introducing much performance degradation. From an extensive set of experiments, we found that the proposed technique achieved promising results under a variety of settings. Future work will address the issues of duplicate human names and explore supervised/semi-supervised learning techniques to further enhance the label quality with affordable human manual refinement efforts.

Acknowledgement

This work was in part supported by Singapore MOE academic tier-1 grant (RG33/11), Microsoft Research Asia

TABLE 5
Evaluation of annotation performance archived by Clustering-based Approximation (CBA)

top T	ORI	ULR	$q_c = 02$		$q_c = 04$		$q_c = 08$		$q_c = 16$	
		$q_c = 01$	BCBA	DCBA	BCBA	DCBA	BCBA	DCBA	BCBA	DCBA
T01	0.574 ± 0.017	0.652 ± 0.019	0.650 ± 0.017	0.644 ± 0.015	0.641 ± 0.016	0.630 ± 0.015	0.638 ± 0.018	0.627 ± 0.015	0.631 ± 0.016	0.624 ± 0.018
T02	0.692 ± 0.015	0.761 ± 0.014	0.758 ± 0.013	0.763 ± 0.015	0.758 ± 0.014	0.765 ± 0.014	0.763 ± 0.014	0.759 ± 0.015	0.767 ± 0.014	0.753 ± 0.013
T03	0.778 ± 0.014	0.820 ± 0.013	0.825 ± 0.013	0.820 ± 0.013	0.828 ± 0.013	0.817 ± 0.012	0.823 ± 0.013	0.819 ± 0.012	0.823 ± 0.011	0.810 ± 0.014
T04	0.813 ± 0.013	0.862 ± 0.013	0.857 ± 0.011	0.855 ± 0.010	0.866 ± 0.011	0.855 ± 0.010	0.855 ± 0.010	0.859 ± 0.011	0.860 ± 0.009	0.852 ± 0.010
T05	0.844 ± 0.012	0.874 ± 0.013	0.873 ± 0.010	0.869 ± 0.009	0.877 ± 0.010	0.874 ± 0.010	0.868 ± 0.010	0.873 ± 0.010	0.876 ± 0.008	0.872 ± 0.010



(a)

(b)

Fig. 9. The label refinement experiment over an artificial dataset. (a) The demo dataset with 9 classes (persons), half of them are mislabeled. (b) The original noisy label matrix and the refined ones achieved by various algorithms. The distances of the refined label matrix to the ideal label matrix (Y_{true}) are shown at the bottom of each figure.

grant, and Singapore National Research Foundation under its Interactive & Digital Media (IDM) Program administrated by Media Development Authority (Research Grant No. MDA/IDM/2012/8/8-2VOL01). Jianke Zhu was supported by National Natural Science Foundation of China under Grants (61103105) and Fundamental Research Funds for the Central Universities.

References

[1] S. C. H. Hoi, J. Luo, S. Boll, D. Xu, and R. Jin, Eds., *Social Media Modeling and Computing*. Springer, 2011. 1

[2] S. Satoh, Y. Nakamura, and T. Kanade, "Name-it: Naming and detecting faces in news videos," *IEEE MultiMedia*, vol. 6, pp. 22–35, 1999. 1

[3] P. T. Pham, T. Tuytelaars, and M.-F. Moens, "Naming people in news videos with label propagation," *Multimedia, IEEE*, vol. 18, no. 3, pp. 44 –55, march 2011. 1

[4] L. Zhang, L. Chen, M. Li, and H. Zhang, "Automated annotation of human faces in family albums," in *ACM Multimedia*, 2003. 1

[5] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y. W. Teh, E. G. Learned-Miller, and D. A. Forsyth, "Names and faces in the news." in *IEEE CVPR*, 2004, pp. 848–854. 1, 2, 3, 8

[6] J. Yang and A. G. Hauptmann, "Naming every individual in news video monologues," in *ACM Multimedia*, 2004, pp. 580–587. 1

[7] J. Zhu, S. C. H. Hoi, and M. R. Lyu, "Face annotation using transductive kernel fisher discriminant," *IEEE Transactions on Multimedia*, vol. 10, no. 1, pp. 86–96, 2008. 1, 2

[8] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. PAMI*, vol. 22, no. 12, pp. 1349–1380, 2000. 1

[9] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu, "Semi-supervised svm batch mode active learning with applications to image retrieval," *ACM TOIS*, vol. 27, pp. 1–29, 2009. 1

[10] X.-J. Wang, L. Zhang, F. Jing, and W.-Y. Ma, "Annosearch: Image auto-annotation by search," in *CVPR*, 2006, pp. 1483–1490. 1, 2

[11] L. Wu, S. C. H. Hoi, R. Jin, J. Zhu, and N. Yu, "Distance metric learning from uncertain side information for automated photo tagging," *ACM TIST*, vol. 2, no. 2, p. 13, 2011. 1

[12] P. Wu, S. C. H. Hoi, P. Zhao, and Y. He, "Mining social images with distance metric learning for automated image tagging," in *Proceedings of the fourth ACM international conference on Web search and data mining*, ser. WSDM '11, Hong Kong, China, 2011, pp. 197–206. 1

[13] D. Wang, S. C. H. Hoi, and Y. He, "Mining weakly labeled web facial images for search-based face annotation," in *SIGIR*, 2011. 2, 3, 8

[14] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE TPAMI*, vol. 19, no. 7, pp. 711 –720, 1997. 2

[15] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Comput. Surv.*, vol. 35, pp. 399–458, 2003. 2

[16] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Tech. Rep. 07-49, 2007. 2

[17] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *ACCV'2010.*, Jun. 2008. 2

[18] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? Metric learning approaches for face identification," in *ICCV*, 2009. 2

[19] Z. Cao, Q. Yin, X. Tang, and J. Sun, "Face recognition with learning-based descriptor," in *CVPR*, 2010, pp. 2707–2714. 2

[20] E. Hjelmås and B. K. Low, "Face detection: A survey," *Computer Vision and Image Understanding*, vol. 83, no. 3, pp. 236–274, 2001. 2

[21] R. Jafri and H. R. Arabnia, "A survey of face recognition techniques," *Journal of Information Processing Systems*, vol. 5, pp. 41–68, 2009. 2

[22] K. Delac and M. Grgic, *Face Recognition*. IN-TECH, 2007. 2

[23] M. G. Kresimir Delac and M. S. Bartlett, *Recent Advances in Face Recognition*. I-Tech Education and Publishing, 2008. 2

[24] A. Hanbury, "A survey of methods for image annotation," *J. Vis. Lang. Comput.*, vol. 19, pp. 617–627, October 2008. 2

[25] Y. Yang, Y. Yang, Z. Huang, H. T. Shen, and F. Nie, “Tag localization with spatial correlations and joint group sparsity,” in *CVPR*, 2011, pp. 881–888. 2

[26] J. Fan, Y. Gao, and H. Luo, “Hierarchical classification for automatic image annotation,” in *ACM SIGIR*, 2007, pp. 111–118. 2

[27] Z. Lin, G. Ding, and J. Wang, “Image annotation based on recommendation model,” in *ACM SIGIR*, 2011, pp. 1097–1098. 2

[28] P. Duygulu, K. Barnard, J. de Freitas, and D. A. Forsyth, “Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary,” in *ECCV*, 2002, pp. 97–112. 2

[29] J. Fan, Y. Gao, and H. Luo, “Multi-level annotation of natural scenes using dominant image components and semantic concepts,” in *ACM Multimedia*, 2004, pp. 540–547. 2

[30] G. Carneiro, A. B. Chan, P. Moreno, and N. Vasconcelos, “Supervised learning of semantic classes for image annotation and retrieval,” *IEEE Tran. PAMI*, pp. 394–410, 2006. 2

[31] C. Wang, F. Jing, L. Zhang, and H.-J. Zhang, “Image annotation refinement using random walk with restarts,” in *ACM Multimedia*, 2006, pp. 647–650. 2

[32] P. Pham, M.-F. Moens, and T. Tuytelaars, “Naming persons in news video with label propagation,” in *VCIDS 2010*, 2010, pp. 1528–1533. 2

[33] J. Tang, R. Hong, S. Yan, T.-S. Chua, G.-J. Qi, and R. Jain, “Image annotation by knn-sparse graph-based label propagation over noisily tagged web images,” *ACM TIST*, vol. 2, pp. 14:1–14:15, 2011. 2

[34] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: Bringing order to the web.” Stanford InfoLab, Technical Report 1999-66, November 1999. 2

[35] X. Rui, M. Li, Z. Li, W.-Y. Ma, and N. Yu, “Bipartite graph reinforcement model for web image annotation,” in *ACM Multimedia*. ACM, 2007, pp. 585–594. 2

[36] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, “Labelme: A database and web-based tool for image annotation,” *Int. J. Comput. Vision*, vol. 77, no. 1-3, pp. 157–173, 2008. 2

[37] Y. Tian, W. Liu, R. Xiao, F. Wen, and X. Tang, “A face annotation framework with partial clustering and interactive labeling,” in *CVPR*. IEEE Computer Society, 2007. 2

[38] J. Cui, F. Wen, R. Xiao, Y. Tian, and X. Tang, “Easyalbum: an interactive photo annotation system based on face clustering and re-ranking,” in *CHI*, 2007, pp. 367–376. 2

[39] D. Anguelov, K. chih Lee, S. B. Göktürk, and B. Sumengen, “Contextual identity recognition in personal photo albums,” in *CVPR2007*, 2007. 2

[40] J. Y. Choi, W. D. Neve, K. N. Plataniotis, and Y. M. Ro, “Collaborative face recognition for improved face annotation in personal photo collections shared on online social networks,” *IEEE Transactions on Multimedia*, vol. 13, no. 1, pp. 14–28, 2011. 2

[41] D. Ozkan and P. Duygulu, “A graph based approach for naming faces in news photos,” in *CVPR*, 2006, pp. 1477–1482. 2

[42] D.-D. Le and S. Satoh, “Unsupervised face annotation by mining the web,” in *ICDM*. IEEE Computer Society, 2008, pp. 383–392. 2

[43] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, “Automatic face naming with caption-based supervision,” in *CVPR*, 2008. 2

[44] ——, “Face recognition from caption-based supervision,” *International Journal of Computer Vision*, 2011. 2

[45] T. Mensink and J. J. Verbeek, “Improving people search using query expansions,” in *ECCV (2)*, 2008, pp. 86–99. 2

[46] T. L. Berg, A. C. Berg, J. Edwards, and D. Forsyth, “Who’s in the picture,” in *NIPS*, 2005. 2

[47] Z. Wu, Q. Ke, J. Sun, and H.-Y. Shum, “Scalable face image retrieval with identity-based quantization and multi-reference re-ranking,” in *CVPR*, 2010, pp. 3469–3476. 2, 10

[48] M. Zhao, J. Yagnik, H. Adam, and D. Bau, “Large scale learning and recognition of faces in web videos,” in *FG*, 2008, pp. 1–7. 3, 8

[49] D. Wang, S. C. H. Hoi, Y. He, and J. Zhu, “Retrieval-based face annotation by weak label regularized local coordinate coding,” in *ACM Multimedia*, 2011, pp. 353–362. 3

[50] D. Wang, S. C. H. Hoi, and Y. He, “A unified learning framework for auto face annotation by mining web facial images,” in *CIKM*, 2012, pp. 1392–1401. 3

[51] X. Zhu, Z. Ghahramani, and J. D. Lafferty, “Semi-supervised learning using gaussian fields and harmonic functions,” in *ICML*, 2003, pp. 912–919. 3

[52] Y.-Y. Sun, Y. Zhang, and Z.-H. Zhou, “Multi-label learning with weak label,” in *AAAI*, M. Fox and D. Poole, Eds. AAAI Press, 2010. 3

[53] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006. 3

[54] J. Zhu, S. C. H. Hoi, and L. V. Gool, “Unsupervised face alignment by robust nonrigid mapping,” in *ICCV*, 2009. 3, 8, 9

[55] C. Siagian and L. Itti, “Rapid biologically-inspired scene classification using features shared with visual attention,” *IEEE TPAMI*, vol. 29, pp. 300–312, 2007. 3

[56] W. Dong, Z. Wang, W. Josephson, M. Charikar, and K. Li, “Modeling lsh for performance tuning,” in *CIKM*, 2008, pp. 669–678. 3

[57] Y. Zhou, R. Jin, and S. C.-H. Hoi, “Exclusive lasso for multi-task feature selection,” in *AISTATS2010*, 2010, pp. 988–995. 4

[58] J. Liu and J. Ye, “Efficient euclidean projections in linear time,” in *ICML*, Montreal, Quebec, Canada, 2009, pp. 657–664. 6

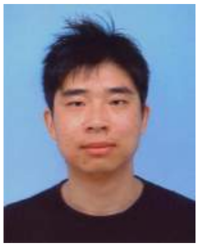
[59] F. Zang and J.-S. Zhang, “Label propagation through sparse neighborhood and its applications,” *Neurocomputing*, 2012. 8

[60] T. Ahonen, A. Hadid, and M. Pietikainen, “Face recognition with local binary patterns,” in *ECCV*, vol. 1, 2004, pp. 469–481. 9

Dayong Wang is currently a PhD candidate in the School of Computer Engineering at the Nanyang Technological University, Singapore. He received his bachelor degree from Tsinghua University, Beijing, P.R. China, in 2008. His research interests are statistical machine learning, pattern recognition, and multimedia information retrieval.

Steven C. H. Hoi is currently an Assistant Professor in the School of Computer Engineering, Nanyang Technological University, Singapore. He received his Bachelor degree in Computer Science from Tsinghua University, Beijing, P.R. China, and his Master and Ph.D degrees in Computer Science and Engineering from Chinese University of Hong Kong. His research interests include machine learning, multimedia information retrieval, web search and data mining. He is a member of IEEE and ACM.

Ying He received the BS and MS degrees in Electrical Engineering from Tsinghua University, and the PhD degree in Computer Science from Stony Brook University. He is currently an associate professor at the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests fall in the broad area of visual computing. He is particularly interested in the problems that require geometric computation and analysis.



Jianke Zhu is currently an Associate Professor at Zhejiang University. He obtained Bachelor degree from Beijing University of Chemical Technology in 2001, his Master degree from University of Macau in 2005, and his PhD degree in the Computer Science and Engineering department at the Chinese University of Hong Kong. His research interests are in pattern recognition, computer vision, and statistical machine learning.