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

Dayong Wang, Steven C.H. Hoi, Ying He
School of Computer Engineering, Nanyang Technological University, Singapore
{s090023, chhoi, yhe}@ntu.edu.sg

ABSTRACT

In this paper, we investigate a search-based face annotation framework by mining weakly labeled facial images that are freely available on the internet. A key component of such a search-based annotation paradigm is to build a database of facial images with accurate labels. This is however challenging since facial images on the WWW are often noisy and incomplete. To improve the label quality of raw web facial images, we propose an effective Unsupervised Label Refinement (ULR) approach for refining the labels of web facial images by exploring machine learning techniques. We develop effective optimization algorithms to solve the large-scale learning tasks efficiently, and conduct an extensive empirical study on a web facial image database with 400 persons and 40,000 web facial images. Encouraging results showed that the proposed ULR technique can significantly boost the performance of the promising search-based face annotation scheme.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Experimentation

Keywords

web facial images, auto face annotation, unsupervised learning

1. INTRODUCTION

Due to the popularity of various digital cameras and the rapid growth of Internet-based photo sharing, recent years have witnessed an explosion of the number of photos captured and stored by consumers. A large collection of photo images which are usually unlabeled raises a great challenge for end users to browse and search. One possible solution is to tag images manually, which is however time-consuming and often costly for large photo collections.

Instead of annotating images manually, another more salient technique to overcome this challenge is *automated image annotation* [7], which aims to automatically assign an image some metadata typically in the form of captions or keywords to describe the semantic concepts/objects in the image. Despite being studied extensively [4, 7, 8], existing techniques for generic image annotation remain far from practical and satisfactory in real-world applications. Unlike the existing generic image annotation, in this paper, we address a specific sub-topic, i.e., *auto face annotation*, which aims to detect human faces from a photo image and to annotate the human names to the facial image.

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 [17].

Conventional face annotation methods usually adapt existing face recognition techniques by training multi-class face classification models from a collection of human-labeled facial images using supervised machine learning techniques [2, 25]. We refer to such kind of conventional techniques as "model-based face annotation". Such approach is however 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.

To address the above limitations, in this paper, we investigate a promising search-based framework for auto face annotation, which aims to exploit large amount of weakly labeled facial images that are freely available on the World Wide Web (WWW). Unlike the conventional model-based approaches, the proposed framework is data-driven and model-free, which annotates a novel facial image by a retrieval-based annotation process [22]. In particular, given a novel facial image for annotation, we first retrieve a set of k most similar facial images from a weakly labeled facial image database, and then label the novel facial image by performing voting on the labels associated with the top k similar facial images.

A key challenge in the above search-based face annotation framework is that labels with web facial images are usually noisy and sometimes may be incomplete due to the nature of image uploading and tagging by WWW users. This is critical as the label quality of the database can considerably affect the final annotation performance of the search-based face annotation process. To overcome this challenge, in this paper, we propose a novel unsupervised label refinement scheme by studying machine learning techniques to enhance the labels purely from the weakly-labeled data without hu-

SIGIR'11, July 24–28, 2011, Beijing, China.
Copyright 2011 ACM 978-1-4503-0757-4/11/07 ...$10.00.

man manual efforts. As a summary, the main contributions in this paper include the following:

- We investigate a promising search-based framework for auto face annotation 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 have implemented the proposed search-based face annotation system and conducted an extensive set of experiments, in which encouraging results were obtained.

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 by learning from weakly labeled data. Section 5 discusses our experiments for performance evaluation, and Section 6 concludes this paper.

2. RELATED WORK

Our work is related to several groups of research.

The first group is on the topics of face detection, verification and recognition. Face detection, verification and recognition are classical research problems in computer vision and pattern recognition, which have been extensively studied for many years [1, 28]. Researchers have developed a variety of face databases for the benchmark of face detection and recognition techniques, such as the well-known FERET database [15]. The traditional studies are often limited for the high-quality databases collected in well-controlled environments. 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 [11, 3]. A comprehensive survey on face detection and recognition topics can be found in [28, 10].

The second group of related work is on the topic of face annotation. In general, face annotation can be viewed as an extended face detection and recognition problem. Some studies in literature have attempted to adapt existing face recognition techniques for face annotation tasks by formulating the problem as a supervised face classification task [2, 26, 25, 17]. For example, the work in [2] adapted the Fisher's linear discriminant analysis method for face annotation, the study in [26] employed a Bayesian method for face annotation, and the work in [25] adopted the Support Vector Machines (SVM) to train and predict the probabilities of human names towards transcript matching faces in the videos. Besides the supervised learning approaches, some existing work also attempts to apply semi-supervised learning for face annotation. For example, the work in [31] proposed a transductive kernel Fisher discriminant for face annotation, which employs both labeled and unlabeled data to train classification models for the annotation tasks. Our work differs from the above existing studies in that our method is model-free by adopting the emerging search-based annotation paradigm for auto face annotation, which is fully data-driven by mining weakly labeled web facial images.

The third group of related work is on the topics of auto image annotation [12, 9, 21] and some recent emerging studies on search-based image annotation [22, 23, 24]. Conventional image annotation approaches have been studied extensively, which usually apply some existing object recognition techniques to train classification models from human-labeled training images [7, 8, 4]. Recently, there is a surge of interests for exploring web image repositories for auto image annotation and object recognition problems using the retrieval-based annotation paradigm [22]. For example, Russell et al. [16] developed a large collection of web images with ground truth labels to facilitate object recognition research. Wang et al. [22] proposed a fast retrieval-based approach for image annotation by studying some efficient hashing technique. Torralba et al. [20] suggested efficient image search and scene matching techniques for exploring a large-scale web image repository. These studies usually concerned more on efficient indexing and searching techniques, while our work focuses on improving the label quality by machine learning techniques.

The final group of closely related work is about mining web facial images, which aims to leverage noisy web facial images for face recognition [2, 13, 27]. For example, Berg et al. [2] crawled a large number of news pictures and captions from the WWW, and proposed a modified k-means clustering approach for cleaning up noisy web facial images. Recently, Le et al. [13] proposed a two-step re-ranking scheme to purify text-based retrieval results for some special names. Zhao et al. [27] proposed a consistency learning method to train face models for famous people 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 that were not designed to optimize the search-based face annotation paradigm, our novel unsupervised label refinement scheme is proposed to optimize the label quality for the search-based face annotation task. Finally, we note that our learning methodology for solving the unsupervised label refinement task is partially inspired by some existing studies in machine learning, including graph-based semi-supervised learning and multi-label learning techniques [32, 19, 5].

3. SEARCH-BASED FACE ANNOTATION FRAMEWORK

Figure 1 illustrates the system flow of the proposed search-based face annotation scheme, which consists of the following steps: (1) facial images data collection, (2) face detection and facial feature extraction, (3) high-dimensional facial feature indexing, (4) learning with weakly labeled data, (5) similar face retrieval, (6) face annotation by majority voting from similar faces with their improved labels. The first four steps are 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 thus should be done very efficiently. We describe each step briefly as follows.

The first step is the collection of facial image data as shown in Figure 1(a), in which we crawled facial images from the WWW by web search engines (e.g., Google) based on a name list that stores the names of persons to be collected. This crawling process produces a collection of facial images, each of them is associated with some human name. 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 region detection and alignment, face region extraction, and facial feature representation. For facial region detection and alignment, we adopt the unsupervised face alignment technique in [30]. For facial feature representation, we extract the GIST features [18] to represent the extracted faces. As a result, each face can be represented as a d-dimensional vector.

The third step of the framework 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 Hash-

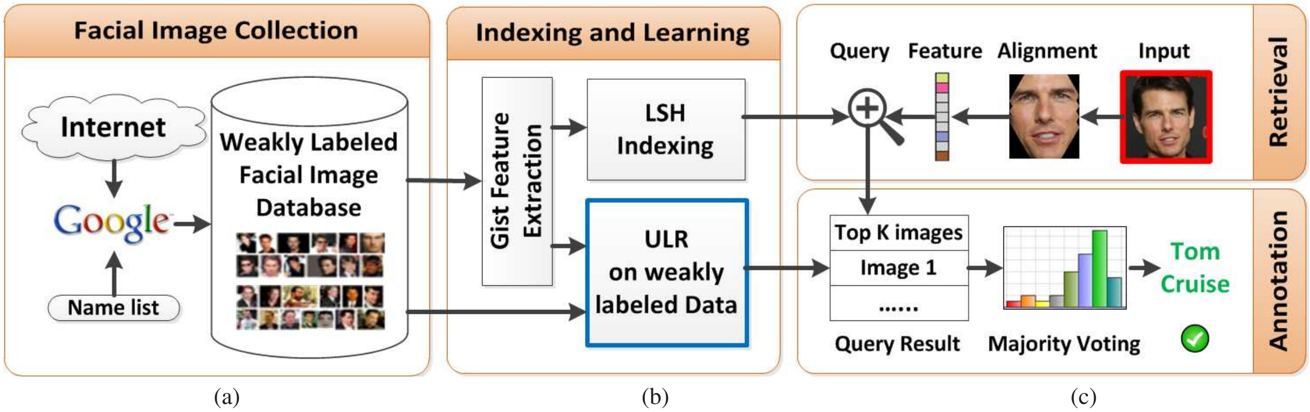


Figure 1: The system flow of the proposed search-based face annotation scheme. (a) We collect weakly-labeled facial images from WWW using web search engines; (b) We perform face detection and alignment, extract GIST features for the detected faces, and finally apply LSH to index the high-dimensional facial features; after that, we apply the proposed Unsupervised Label Refinement (ULR) method to refine the labels of the facial images; (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.

ing (LSH) [6], a popular and effective high-dimensional indexing technique for approximate nearest neighbor search.

Besides the indexing step, another key step of our framework is to engage an unsupervised learning scheme to enhance label quality of the weakly labeled facial images. This process is critical to the entire search-based annotation framework since the label quality considerably affects the final annotation performance.

All the above are the processing steps 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 pay our main attention on the 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 ON WEAKLY LABELED FACIAL IMAGES

In this section, we present a novel Unsupervised Label Refinement (ULR) scheme to refine the labels of web facial image data by learning with weakly labeled data. In the following, we first introduce some preliminaries and notations followed by the problem formulation and the proposed algorithms.

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 $\Omega = \{n_1, n_2, \ldots, n_m\}$ 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.

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

$$W_{ij} = \begin{cases} e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}} & \text{if } \mathbf{x}_i \in \mathfrak{n}_K(\mathbf{x}_j) || \mathbf{x}_j \in \mathfrak{n}_K(\mathbf{x}_i) \\ 0 & otherwise \end{cases} \quad (1)$$

where $\mathfrak{n}_K(\mathbf{x}_j)$ denotes the $K \ll n$ nearest neighbor list of the data point $\mathbf{x}_j$ based on Euclidean distance.

4.2 Problem Formulation

The goal of the Unsupervised Label Refinement (ULR) task is to learn a refined label matrix $F^* \in \mathbb{R}^{n \times m}$ to improve the initial raw label matrix Y. This is challenging since we have nothing else but the raw label matrix Y and the data examples X themselves. To attack this challenge, 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 $E_s(F, W)$:

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

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

Directly optimizing the above loss function is problematic as it will yield a trivial solution. To tackle this challenge, 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 the new label matrix F, we shall avoid the solution F being deviated too much from the raw label matrix Y. To this end, we formulate the following optimization task for the unsupervised label refinement problem by including a regularization term $E_p(F,Y)$ to reflect this concern:

$$F^* = \arg\min_{F\geq 0} E_s(F,W) + \alpha \cdot E_p(F,Y) \tag{3}$$

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 $E_p(F,Y)$:

$$E_p(F,Y) = \|(F-Y)\circ S\|_F^2 \tag{4}$$

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 [29]:

$$E_e(F) = \sum_{i=1}^{n} (\|F_{i*}\|_1)^2 \tag{5}$$

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:

$$F^* = \arg\min_{F\geq 0} g(F) \tag{6}$$

$$g(F) = E_s(F,W) + \alpha E_p(F,Y) + \beta E_e(F) \tag{7}$$

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:

$$F^* = \arg\min_{F\geq 0} E_s(F,W) + \alpha E_p(F,Y) \tag{8}$$

$$s.t. \quad \|F_{i*}\|_1 \leq \varepsilon, i = 1,\ldots,n \tag{9}$$

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 of totally 40,000 facial images in our experiment, F is a 40000×400 matrix that consists of 16 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.6, 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:

$$g(F) = tr(F^\top LF) + \alpha\|(F-Y)\circ S\|_F^2 + \beta\|F\cdot\mathbf{1}\|_F^2 \tag{10}$$

$$= \tilde{\mathbf{f}}^\top Q\tilde{\mathbf{f}} + \mathbf{c}^\top\tilde{\mathbf{f}} + \mathbf{h} \tag{11}$$

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 $k \times k$.

The above optimization is clearly a QP problem. To solve it efficiently, we adopt 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:

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} q(\mathbf{x}|Q,\mathbf{c}) = \mathbf{x}^\top Q\mathbf{x} + \mathbf{c}^\top\mathbf{x} \quad \mathbf{s.t.}\ \mathbf{x} \geq 0 \tag{12}$$

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

$$p_t(\mathbf{x},\mathbf{z}) = q(\mathbf{z}) + < \mathbf{x}-\mathbf{z}, \nabla q(\mathbf{z}) > + \frac{t}{2}\|\mathbf{x}-\mathbf{z}\|_F^2 \tag{13}$$

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 constructed combination of the two previous approximate solutions $\mathbf{x}^{(k-1)}$ and $\mathbf{x}^{(k-2)}$. The approximation $\mathbf{x}^{(k)}$ is achieved by the following optimization:

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}}\ p_t(\mathbf{x},\mathbf{z}^{(k)}) \quad \mathbf{s.t.}\ \mathbf{x} \geq 0 \tag{14}$$

After ignoring terms that do not depend on $\mathbf{x}$, the former optimization problem Eq.14 could be equally presented as:

$$\min_{\mathbf{x}\geq\mathbf{0}}\ \mathbf{g}^\top\mathbf{x} + \frac{t}{2}\|\mathbf{x}-\mathbf{z}^{(k)}\|^2 = t\sum_i[\frac{1}{2}(x_i - z_i^{(k)})^2 + \frac{g_i}{t}x_i] \tag{15}$$

where $\mathbf{g} = 2Q\mathbf{z}^{(k)} + \mathbf{c}$. The closed-form solution is given below:

$$x_i = \max(z_i^{(k)} - g_i/t, 0) \tag{16}$$

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: $\mathbf{x}^\star \in \mathbb{R}^{n\cdot m}$
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. 14;
        Case CCF : Achieve $\mathbf{x}^{(k)}$ with Eq. 19;
        $\alpha_k = \frac{1+\sqrt{4\alpha_{k-1}^2+1}}{2}$;
        $\mathbf{z}^{(k)} = \mathbf{x}^{(k)} + \frac{\alpha_{k-1}-1}{\alpha_k}(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})$;
        $k = k + 1$;
    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 very large-scale problems.

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

$$F_{i*}^{(t+1)} = \arg\min_{\mathbf{f}^\top} \ \Psi(\mathbf{f} \mid F^{(t)}, i) \quad \textbf{s.t. } \mathbf{f} \geq 0 \tag{17}$$

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.17 is also a smooth QP problem, but much smaller than the original Eq.10. 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. 17;
            Case CCF: Achieve $F_{i*}^{(t+1)}$ with Eq. 25;
        $t = t + 1$;
    until CONVERGENCE;
```

4.3.2 Algorithm for Convex-Constraint Formulation

For the convex-constraint formulation, by doing vectorization, we can reformulate Eq. 8 into the following:

$$\min_{\mathbf{x}\geq 0} \mathbf{x}^\top Q^\dagger\mathbf{x} + \mathbf{c}^\top\mathbf{x} \quad \textbf{s.t.} \sum_{k=0}^{m-1} x_{k\cdot n+i} \leq \varepsilon, i = 1,\ldots,n. \tag{18}$$

where $Q^\dagger = U + \alpha R$, $\varepsilon \geq 1$, and all the other symbols are the same as Eq. 10. We also apply the multi-step gradient scheme to solve Eq. 18, however the constraint for the sub-problem is slightly different from Eq. 15, which is defined:

$$\min_{\mathbf{x}\geq 0} \ \frac{t^\dagger}{2}\|\mathbf{x} - \mathbf{v}\|^2 \quad \textbf{s.t.} \sum_{k=0}^{m-1} x_{k\cdot n+i} \leq \varepsilon, i = 1\ldots,n \tag{19}$$

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 the vector $\mathbf{x}$ 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. 19 could be reformulated as:

$$\begin{aligned}\min_{\bar{\mathbf{x}}^0,\bar{\mathbf{x}}^1,\ldots,\bar{\mathbf{x}}^n} \quad & \frac{t^\dagger}{2}\sum_{i=1}^{n}\|\bar{\mathbf{x}}^i - \bar{\mathbf{v}}^i\|^2 \\ \textbf{s.t.} \quad & \|\bar{\mathbf{x}}^i\|_1 \leq \varepsilon,\ \bar{\mathbf{x}}^i \geq 0,\ i = 1,\ldots,n.\end{aligned} \tag{20}$$

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 [14]. Specifically, we can obtain the optimal solution $\bar{\mathbf{x}}^{i\star}$ for $\bar{\mathbf{x}}^i$ with the following problem:

$$\bar{\mathbf{x}}^{i\star} = \arg\min_{\bar{\mathbf{x}}^i} \|\bar{\mathbf{x}}^i - \bar{\mathbf{v}}^i\|^2 \quad \textbf{s.t.} \quad \|\bar{\mathbf{x}}^i\| \leq \varepsilon; \bar{\mathbf{x}}^i \geq 0. \tag{21}$$

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

$$\bar{x}_j^{i\star} = sign(\bar{v}_j^i) \times \max(|\bar{v}_j^i| - \lambda^\star, 0), j = 1,2,...m. \tag{22}$$

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

$$\lambda^\star = \begin{cases} 0 & \sum_{k\in S}|\bar{v}_k^i| \leq \varepsilon, \\ \bar{\lambda} & \sum_{k\in S}|\bar{v}_k^i| > \varepsilon. \end{cases} \tag{23}$$

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

$$f(\lambda) = \sum_{k\in S}\max(|\bar{v}_k^i| - \lambda, 0) - \varepsilon. \tag{24}$$

$f(\lambda)$ is continuous and monotonically decreasing in $(-\infty,\infty)$. The root $\bar{\lambda}$ can be obtained by the bisection search algorithm in linear time. An improved searching scheme was also proposed in [14] using the characteristic of function $f(\lambda)$.

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. 17:

$$F_{i*}^{(t+1)} = \arg\min_{\mathbf{f}^\top} \ \Psi^\dagger(\mathbf{f} \mid F^{(t)}, i) \quad \textbf{s.t.} \ \|\mathbf{f}\|_1 \leq \varepsilon, \mathbf{f} \geq 0 \tag{25}$$

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. 17 except $\hat{Q}^\dagger = L_{ii}I_M + \alpha R$. Eq. 25 is a special case of the optimization problem in Eq. 18, 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.

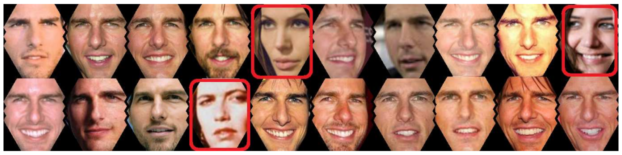
5. EXPERIMENTS

In this section, we conducted extensive experiments to evaluate the performance of the proposed ULR technique for automated face annotation. In the following, we first introduce our experimental testbed, then discuss the comparison schemes and experimental setup, and finally present our experimental results.

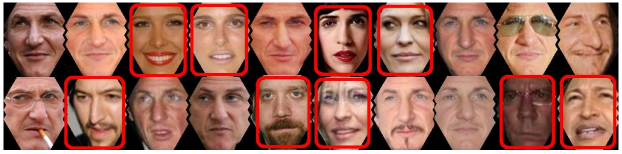
5.1 Experiment Testbed

To build our testbed, we first collected a name list consisting of 400 popular actor and actress names downloaded from the **IMDb** website http://www.imdb.com. We collected those names with the billboard: "Most Popular People Born In **yyyy**" of **IMDb**, where **yyyy** is the born year, e.g., the webpage [1] 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. We submitted each name from the list as a query to search for related web images by Google image search engine. The top 200 retrieved web images are crawled automatically. After that we use the OpenCV toolbox to detect the faces and adopt

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



(a)Tom Cruise



(b)Sean Penn

Figure 2: Two example sets of weakly labeled facial images where wrongly labeled images were highlighted by red boxes: (a) a good case of most images correctly labeled, and (b) a bad case of half of them wrongly labeled.

the DLK algorithm [30] to align facial images into the same well-defined position. The no-face-detected web images were ignored. As a result, we collected over $40,000$ 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. Figure 2 shows two examples in our database.

We also built a "test dataset" by randomly choosing 80 names from our name list. We submitted each selected name as a query to Google and crawled about 200 images from top 200 to 400 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.

We run all the experiments on a PC with an Intel(R) Xeon(R) CPU (W3520), 12G memory and MATLAB 2010(b). To handle a very large-scale optimization task of 16-million unknown variables, we adopted the *Parallel Toolbox* in Matlab to exploit the parallel computation power.

5.2 Comparison Schemes and Setup

In our experiments, we implemented all the proposed algorithms for solving the 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 [27] proposed to enhance the weakly labeled facial image database, denoted as "CL" for short.
- "MKM": a modified K-means clustering algorithm [2] proposed to cluster the image database and clean up noise images, denoted as "MKM" for short.
- "ULR": the proposed unsupervised label refinement method, denoted as "ULR" for short.

For a fair comparison to the above approaches, we adopted the same GIST features to represent the facial images. 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 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 Auto Face Annotation

Table 1 and Figure 3 show the average annotation performance (hit rates) at different T values, in which both mean and standard deviation were reported. Several observations can be drawn from the results.

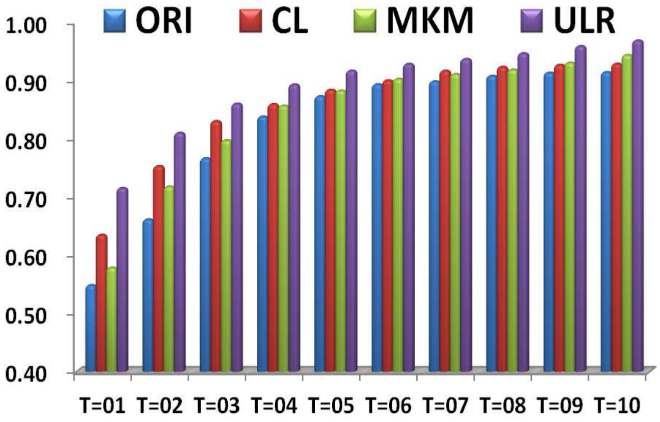


Figure 3: Evaluation of auto face annotation performance in terms of *hit rates* at top T annotated names.

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 and MKM algorithm. For example, by comparing the hit rate at the first annotated name, ORI, CL and MKM achieved 54.8%, 63.4% and 57.8% respectively, while ULR can significantly boost the hit rate to 71.5%. For the hit rate at top 10 annotated names, the result of ORI, CL and MKM are 91.4% and 92.8% and 94.3% respectively, while ULR can achive a better hit rate 96.8%. The promising result validates the effectiveness of the proposed ULR technique for improving search-based face annotation. Second, when T is small, the hit rate gap, i.e., the hit rate difference between ORI and ULR is more significant, which verifies that the proposed ULR algorithm could efficiently refine the noisy data. Finally, we observed that the annotation performance increases slowly when $T > 5$. In practice, we usually focused on the a small T value since users typically would not be interested in a long list of annotated names.

Table 1: 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	T=06	T=07	T=08	T=09	T=10
ORI	0.548 ± 0.013	0.661 ± 0.011	0.766 ± 0.009	0.837 ± 0.010	0.872 ± 0.010	0.892 ± 0.009	0.898 ± 0.009	0.907 ± 0.010	0.912 ± 0.010	0.914 ± 0.010
CL	0.634 ± 0.012	0.752 ± 0.010	0.829 ± 0.010	0.858 ± 0.010	0.883 ± 0.009	0.899 ± 0.009	0.916 ± 0.009	0.922 ± 0.008	0.926 ± 0.007	0.928 ± 0.007
MKM	0.578 ± 0.011	0.717 ± 0.012	0.797 ± 0.012	0.856 ± 0.008	0.882 ± 0.010	0.902 ± 0.009	0.910 ± 0.009	0.918 ± 0.009	0.930 ± 0.009	0.943 ± 0.008
ULR	0.715 ± 0.008	0.809 ± 0.005	0.859 ± 0.009	0.892 ± 0.007	0.916 ± 0.010	0.927 ± 0.007	0.936 ± 0.007	0.946 ± 0.006	0.958 ± 0.007	0.968 ± 0.007

5.4 Evaluation on Varied Top-K Retrieved Images and Top-T Annotated Names

This experiment aims to examine the annotation performance under 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 performance differences between the ULR algorithm and the baseline algorithm as well as the CL algorithm are mostly similar to the observations in the previous experiment. The face annotation performance of varied K and T values are illustrated in Figure 4 and Table 2 where both mean and standard deviation results were reported.

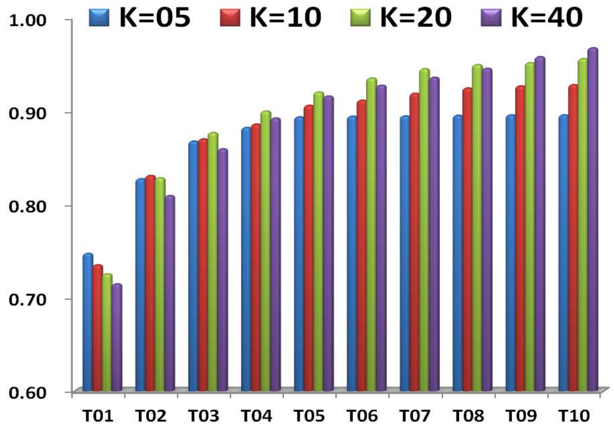


Figure 4: Annotation performance of varied K and T values.

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.

Table 2: Annotation performance of varied K and T values.

	T=01	T=02	T=03	T=04	T=05	T=06	T=07	T=08	T=09	T=10
K=05	0.747 ± 0.014	0.827 ± 0.012	0.868 ± 0.009	0.882 ± 0.009	0.894 ± 0.008	0.894 ± 0.008	0.895 ± 0.008	0.895 ± 0.008	0.896 ± 0.008	0.896 ± 0.008
K=10	0.735 ± 0.011	0.831 ± 0.013	0.870 ± 0.013	0.886 ± 0.012	0.906 ± 0.010	0.911 ± 0.010	0.919 ± 0.010	0.924 ± 0.010	0.927 ± 0.010	0.928 ± 0.009
K=20	0.725 ± 0.011	0.828 ± 0.010	0.877 ± 0.012	0.900 ± 0.010	0.920 ± 0.011	0.935 ± 0.007	0.945 ± 0.007	0.950 ± 0.006	0.952 ± 0.006	0.956 ± 0.006
K=40	0.715 ± 0.013	0.809 ± 0.013	0.859 ± 0.012	0.892 ± 0.010	0.916 ± 0.009	0.927 ± 0.008	0.936 ± 0.008	0.946 ± 0.006	0.958 ± 0.005	0.968 ± 0.007

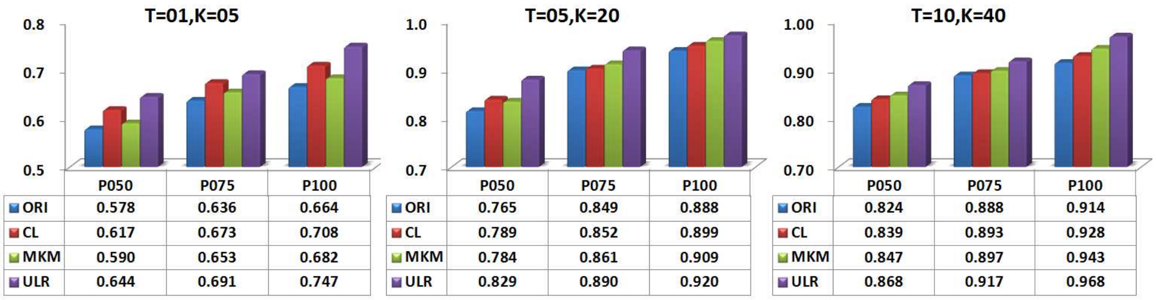
5.5 Evaluation on Varied Numbers of Images per Person in Database

This experiment aims to further examine how the annotation performance is affected by 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 variants 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.

Figure 5 shows the experiment results of average annotation performance. We can draw some observations. First of all, 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. Second, similar to the previous observations, ULR consistently boosts the annotation performance for all the databases, and achieves the best performance by beating the other competitors for all the cases. Finally, we 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.

5.6 Evaluation of Parameter Sensitivity

There are two key parameters α and β in the ULR algorithm. In the previous experiments, we found the best values by cross validation. In this experiment, we further examine their sensitivity to the annotation performance. Figure 6 shows an evaluation of annotation performance by a grid search on varied values of parameter $\alpha \in [0.1, 0.6]$ and $\beta \in [0, 0.7]$ in one of cross validation experiments. The value of each vertex on the color mesh represents the hit rate gap between ULR and ORI.



	P050	P075	P100
ORI	0.578	0.636	0.664
CL	0.617	0.673	0.708
MKM	0.590	0.653	0.682
ULR	0.644	0.691	0.747

	P050	P075	P100
ORI	0.765	0.849	0.888
CL	0.789	0.852	0.899
MKM	0.784	0.861	0.909
ULR	0.829	0.890	0.920

	P050	P075	P100
ORI	0.824	0.888	0.914
CL	0.839	0.893	0.928
MKM	0.847	0.897	0.943
ULR	0.868	0.917	0.968

Figure 5: The annotation performance on three different databases, which have different numbers of images per person. Specifically, P050, P075, and P100 denote the databases having the top 50, 75, and 100 retrieval images per person, respectively.

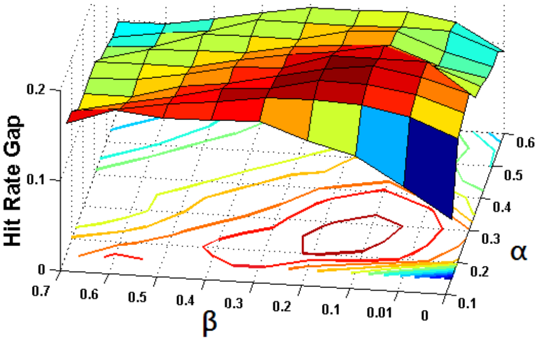


Figure 6: Evaluation of parameter sensitivity with respect to α and β for $T = 1$. The values of the vertex on the color mesh illustrate the hit rate gap between ORI and ULR.

Some observations can be drawn from the results. First, we found that for all α and β values, the proposed ULR scheme always has a positive improvement over ORI with original labels. Second, we found that the best α and β values are about 0.3 and 0.1, respectively. Further, we noticed that it is not difficult to find such good α and β values to get comparable results. Typically, ULR attains fairly good results when choosing $\alpha \in [0.2, 0.4]$ and $\beta \in [0.01, 0.4]$. These results show that ULR is quite robust to the parameters and could always enhance the annotation performance.

5.7 Evaluation of Optimization Efficiency

This section aims to conduct extensive evaluations on the running time cost by the four proposed algorithms. To distinguish the previous four algorithms clearly, in the following subsections, we will refer to the four algorithms by 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.

5.7.1 *Time Cost Evaluation on Artificial Data*

We first compare 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 adopted 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 = 2000, 4000, 6000, 8000, 10000. The goal of the ULR optimization is to optimize the refined label matrix $F \in \mathbb{R}^{n\times m}$, which has the total number of unknown variables V = 40000, 160000, 360000, 640000, 1000000, respectively for each of the above cases. For the iteration number, we set it to 50 for both algorithms.

We randomly generated the artificial datasets and run the algorithms over these random datasets. This procedure was repeated five times. Table 3 show the average running time cost, where the first two columns are the means and their standard deviations obtained by both SRF-MGA and CCF-MGA algorithms, respectively. Figure 7 (a) further illustrates these results. Some observations can be drawn from these results as follows.

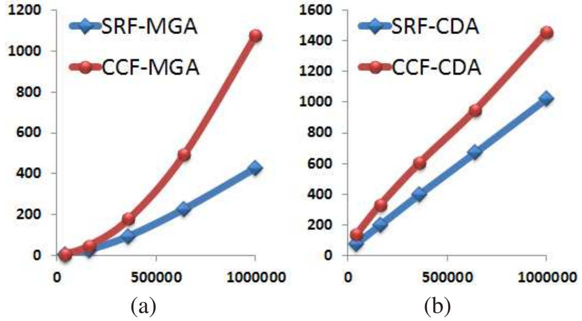


Figure 7: The running time of the proposed four algorithm (SRF-MGA, CCF-MGA, SRF-CDA, CCF-CDA). The group size P is 100. The x-axis presents the number of variables V, The y-axis is the running time(seconds).

First of all, it is clear that increasing the number of variables leads to the increase of the running time cost for both algorithms. Second, by comparing the two algorithms based on two different formulations, we found 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 w.r.t 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 using the data shown in Figure 7(a), 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.

We compare running time cost of SRF-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

Table 3: Average running time (seconds) of the proposed algorithms, where the 2nd rows show the standard deviations.

V	SRF-MGA	CCF-MGA	SRF-CDA	CCF-CDA
40000	4.80	6.76	76.63	136.50
	± 0.15	± 0.06	± 0.07	± 0.09
160000	29.04	49.05	197.92	330.30
	± 0.14	± 0.07	± 0.29	± 37.06
360000	95.43	178.45	399.11	607.20
	± 0.56	± 0.03	± 0.13	± 12.01
640000	228.44	494.29	670.23	950.40
	± 0.23	± 8.60	± 4.60	± 26.46
1000000	428.46	1076.04	1022.70	1457.40
	± 0.04	± 11.99	± 8.94	± 17.68

to 30 and fix the inner iteration number w.r.t. their subproblems to 30. The average running time cost and their standard deviations are illustrated in the last two columns of Table 3 and Figure 7(b).

Firstly, the SRF-based algorithm SRF-CDA spent less time cost than the CCF-based algorithm CCF-CDA. Second, unlike the previous results of the MGA based algorithms, we found that the running time cost grows almost linearly w.r.t the number of variables for both CDA based algorithms. More specifically, by fitting the time cost function $T = a \times V^b$ w.r.t. 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 show 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 datasets.

5.7.2 Time Cost Evaluation on Real 400-Person Data

This experiment is to evaluate running time cost of the CDA-based algorithms (SRF-CDA and CCF-CDA) on our 400-person weakly labeled real web facial image dataset. For this real dataset, we have 400 persons and about 100 images for each person, which leads to 40000 images and 16-million unknown variables in our optimization task. We skipped the evaluation of MGA-based algorithms (SRF-MGA and CCF-MGA) since they are computationally too intensive for this large-scale experiment.

We randomly chose $P = \{50, 75, 100\}$ images from each person to build three databases of different sizes. We refer to these databases as $P050$,$P075$ and $P100$, respectively. We set the outer iteration number to 10 for both algorithms, and fixed the inner iteration number for their subproblems to 30. First, we employed the CDA-based algorithms (SRF-CDA and CCF-CDA) directly on those three databases, then we adopted the "*Parallel Computing Toolbox*" in Matlab to speed up the loop structure. We simply used the command "*matlabpool* 4" to estimate 4 local labs for the parallel computing task. The final results are presented in Table 4.

Two observations can be drawn from the above results. First, the same as the previous results, SRF-CDA is faster than CCF-CDA on the real database. Secondly, after applying the "*Parallel Computing Toolbox*", the running time is significantly reduced, which saved roughly two-third of the total time cost.

5.7.3 Convergence Evaluation

This experiment is to evaluate the convergence rates of the four different optimization algorithms. In particular, we consider a toy dataset with $m = 100$, $P = 100$, $n = 10000$. Figure 8 show the evaluations on the objective functions of the four different algorithms. For each algorithm, the average running time per iteration was also displayed on each of the figures. Some observations can be drawn from the results.

Table 4: The running time cost (seconds) of SRF-CDA and CCF-CDA algorithms on 400-person retrieval database. The top two rows show the running time cost of algorithms without parallel computing. The last two rows show the results using"*Parallel Computing Toolbox*" in Matlab.

Sequential	P050	P075	P100
SRF-CDA	2726.1 ± 27.9	4078.0 ± 26.3	5506.0 ± 22.7
CCF-CDA	2999.6 ± 33.4	4519.6 ± 34.7	6131.3 ± 31.8
Parallel	**P050**	**P075**	**P100**
SRF-CDA	1072.3 ± 16.6	1556.0 ± 13.2	2022.6 ± 13.3
CCF-CDA	1184.1 ± 18.6	1805.9 ± 21.5	2337.2 ± 11.3

First of all, it is clear that all the algorithms converge quickly. In particular, the two MGA-based algorithms almost converged after 10 iterations, which is slightly slower than the two CDA-based algorithms that almost converged after 8 iterations. Second, with the same formulation, we can see that the final objective values obtained by two different solvers are very close, which validates the correctness of the proposed algorithms. Finally, in terms of time cost per iteration, we found that the algorithms based the SRF formulation are faster than the algorithms based on the CCF formulations. More details about the running time cost comparison are given in the subsequent section.

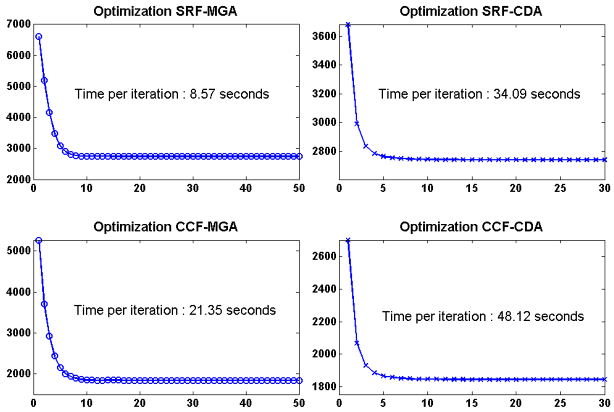


Figure 8: The objective function values of the four different optimization algorithms at each iteration on the toy data with $m = 100$, $P = 100$ and $n = 10000$. The average running time per iteration was also displayed on each of the figures.

6. DISCUSSIONS AND LIMITATIONS

Despite the encouraging results, our work is limited in some aspects. First, in our experiments, we assume each name corresponds to a unique single person, i.e., we do not consider the duplicate name case. This is however possible in reality. Second, we assume the top retrieved web facial images are related to a query human name. This is clearly true for popular human beings, such as movie stars and famous politicians. However, when the query facial image is not a popular person, there may not exist many relevant facial images on the WWW, which is a limitation of all existing data-driven annotation techniques. Third, comparing with the whole WWW, our current facial image database is still not large, though the essential optimization task in our problem is huge. In our future work, we plan to collect a large database, and develop more efficient algorithms to resolve the optimization task.

7. CONCLUSION

This paper investigated a promising search-based face annotation framework for mining weakly labeled facial images freely available on the WWW. To enhance the quality of the noisy and incomplete labels of the web facial images, we proposed a novel Unsupervised Label Refinement (ULR) technique by learning to refine the class labels from the weakly-labeled data. To make the proposed technique feasible for large-scale problems, we presented efficient and scalable optimization algorithms, which successfully solved a ULR optimization task on a real-world web facial image dataset with 400 persons and 40000 facial images by a single PC. From an extensive set of experiments, we found that the proposed technique achieved promising results with about 96% average hit rate of top-10 annotated names over a challenging test set with various web facial images captured in a wild. Our results also indicated that the proposed ULR technique significantly surpassed the other competitors, including the baseline with the original labels and other regular solutions in literature. For the future work, we will further speed up the current solution for very large-scale applications and investigate other machine learning techniques to improve the label refinement task.

Acknowledgement

This work was supported by the Singapore National Research Foundation Interactive Digital Media R&D Program, under research grant NRF2008IDM-IDM004-006.

8. REFERENCES

[1] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. PAMI*, 19(7):711–720, 1997.

[2] 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 *CVPR (2)*, pages 848–854, 2004.

[3] Z. Cao, Q. Yin, X. Tang, and J. Sun. Face recognition with learning-based descriptor. In *CVPR*, pages 2707–2714, San Francisco, CA, 2010.

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

[5] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[6] W. Dong, Z. Wang, W. Josephson, M. Charikar, and K. Li. Modeling lsh for performance tuning. In *CIKM*, 2008.

[7] 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*, pages 97–112, 2002.

[8] J. Fan, Y. Gao, and H. Luo. Multi-level annotation of natural scenes using dominant image components and semantic concepts. In *ACM Multimedia*, pages 540–547, 2004.

[9] J. Fan, Y. Gao, H. Luo, and G. Xu. Automatic image annotation by using concept-sensitive salient objects for image content representation. In *SIGIR*, pages 361–368, 2004.

[10] E. Hjelmås and B. K. Low. Face detection: A survey. *Computer Vision and Image Understanding*, 83(3), 2001.

[11] 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. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[12] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *SIGIR*, pages 119–126, 2003.

[13] D.-D. Le and S. Satoh. Unsupervised face annotation by mining the web. In *IEEE ICDM2008*, pages 383–392, 2008.

[14] J. Liu and J. Ye. Efficient euclidean projections in linear time. In *ICML*, pages 657–664, Montreal, Quebec, Canada, 2009.

[15] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Trans. PAMI*, 22(10):1090–1104, 2000.

[16] 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*, 77(1-3):157–173, 2008.

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

[18] C. Siagian and L. Itti. Rapid biologically-inspired scene classification using features shared with visual attention. *IEEE Trans. PAMI*, 29:300–312, Feb 2007.

[19] Y.-Y. Sun, Y. Zhang, and Z.-H. Zhou. Multi-label learning with weak label. In *AAAI*, 2010.

[20] A. Torralba, Y. Weiss, and R. Fergus. Small codes and large databases of images for object recognition. In *CVPR*, 2008.

[21] C. Wang, L. Zhang, and H.-J. Zhang. Learning to reduce the semantic gap in web image retrieval and annotation. In *SIGIR*, pages 355–362, 2008.

[22] X.-J. Wang, L. Zhang, F. Jing, and W.-Y. Ma. Annosearch: Image auto-annotation by search. In *CVPR*, pages 1483–1490, 2006.

[23] L. Wu, S. C. H. Hoi, R. Jin, J. Zhu, and N. Yu. Distance metric learning from uncertain side information with application to automated photo tagging. In *ACM Multimedia*, pages 135–144, 2009.

[24] P. Wu, S. C. H. Hoi, P. Zhao, and Y. He. Mining social images with distance metric learning for automated image tagging. In *WSDM*, pages 197–206, 2011.

[25] J. Yang and A. G. Hauptmann. Naming every individual in news video monologues. In *ACM Multimedia*, pages 580–587, New York, NY, USA, 2004.

[26] L. Zhang, L. Chen, M. Li, and H. Zhang. Automated annotation of human faces in family albums. In *ACM Multimedia*, pages 355–358, Berkeley, CA, USA, 2003.

[27] M. Zhao, J. Yagnik, H. Adam, and D. Bau. Large scale learning and recognition of faces in web videos. In *FG*, pages 1–7, 2008.

[28] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Comput. Surv.*, 35(4):399–458, 2003.

[29] Y. Zhou, R. Jin, and S. C.-H. Hoi. Exclusive lasso for multi-task feature selection. In *JMLR W&C Proceedings (AISTATS2010)*, pages 988–995, Oct 2010.

[30] J. Zhu, S. C. H. Hoi, and L. V. Gool. Unsupervised face alignment by robust nonrigid mapping. In *ICCV*, 2009.

[31] J. Zhu, S. C. H. Hoi, and M. R. Lyu. Face annotation using transductive kernel fisher discriminant. *IEEE Transactions on Multimedia*, 10(1):86–96, 2008.

[32] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.