Learning to Name Faces: A Multimodal Learning Scheme for Search-Based Face Annotation

Dayong Wang[†], Steven C.H. Hoi[†], Pengcheng Wu[†],
Jianke Zhu[‡], Ying He[†], Chunyan Miao[†]
[†]School of Computer Engineering, Nanyang Technological University, Singapore.
[‡]College of Computer Science, Zhejiang University, Hangzhou, 310027, China.
{s090023, chhoi, wupe0003, yhe, ascymiao}@ntu.edu.sg, jkzhu@zju.edu.cn

ABSTRACT

Automated face annotation aims to automatically detect human faces from a photo and further name the faces with the corresponding human names. In this paper, we tackle this open problem by investigating a search-based face annotation (SBFA) paradigm for mining large amounts of web facial images freely available on the WWW. Given a query facial image for annotation, the idea of SBFA is to first search for top-n similar facial images from a web facial image database and then exploit these top-ranked similar facial images and their weak labels for naming the query facial image. To fully mine those information, this paper proposes a novel framework of Learning to Name Faces (L2NF) – a unified multimodal learning approach for search-based face annotation, which consists of the following major components: (i) we enhance the weak labels of top-ranked similar images by exploiting the "label smoothness" assumption; (ii) we construct the multimodal representations of a facial image by extracting different types of features; (iii) we optimize the distance measure for each type of features using distance metric learning techniques; and finally (iv) we learn the optimal combination of multiple modalities for annotation through a *learning to rank* scheme. We conduct a set of extensive empirical studies on two real-world facial image databases, in which encouraging results show that the proposed algorithms significantly boost the naming accuracy of search-based face annotation task.

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, supervised learning

1. INTRODUCTION

Automated face annotation aims to automatically detect human faces from a photo image and name the facial image with the corresponding human names, which sometimes is termed as "face naming" or "face tagging" in some existing studies. It is an important yet very challenging problem in multimedia information retrieval, which is highly desirable for many real-world applications, such as online photo management or face annotation for video summarization. One possible way is to directly apply some classical face recognition methods [3, 30, 33]. For exmaple, one can apply supervised machine learning techniques to train face classification models from a collection of well-controlled labeled facial images and then apply the models to name a new facial image. However, such kinds of "*model-based face annotation*" techniques suffer from some common drawbacks, e.g., being difficult and expensive to collect large high-quality training data and being nontrivial for adding new training data.

Recent years have witnessed an emerging promising direction to tackle the automated face annotation challenge, i.e., the "*Search-Based Face Annotation*" (SBFA) paradigm [37, 38] which attempts to explore content-based image retrieval (CBIR) techniques [16, 43] in mining massive WWW facial images freely available on the internet, such as popular social sharing web sites (e.g., Flickr or Facebook). Due to the noisy nature of web images, the raw labels of web facial images are often noisy without extra manual effort, in which some facial images are tagged with incorrect/incomplete names. We refer to such kind of raw facial image database as "weakly labeled web facial image database".

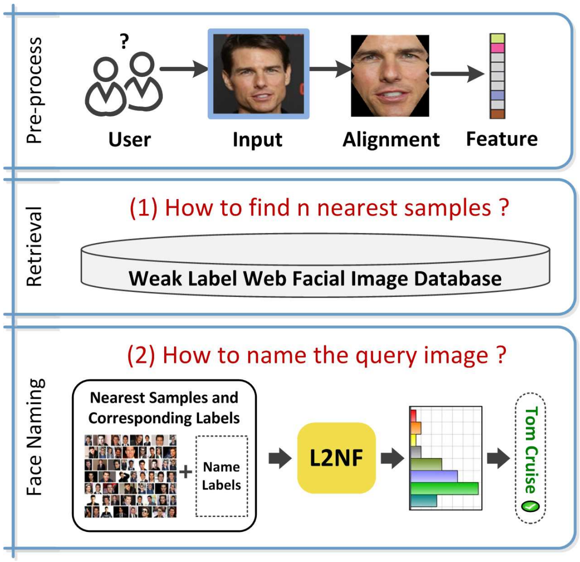

Pre-process
?
User
Input
Alignment
Feature
Retrieval
(1) How to find n nearest samples ?
Weak Label Web Facial Image Database
Face Naming
(2) How to name the query image ?
Nearest Samples and Corresponding Labels
+
Name Labels
L2NF
Tom Cruise

Figure 1: The framework of Search-Based Face Annotation.

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.
Copyright 2013 ACM 978-1-4503-2034-4/13/07 ...$15.00.

Figure 1 illustrates the basic framework of the "*Search-based Face Annotation*" paradigm, which consists of three main stages: (i) given a query facial image, it typically involves a pre-processing stage, including face detection, face alignment, and facial feature extraction; consequently, the input facial image is represented as feature vectors in the facial feature space; (ii) we retrieve the top-n similar instances of the query facial image from the large-scale weakly labeled web facial image database using content-based image retrieval techniques; and (iii) finally, we aim to name the query image by mining the top-ranked similar images and the corresponding weak name labels. Such a paradigm was inspired by the search-based image annotation [39] for generic image annotation as face annotation can be generally viewed as a special case of image annotation [9, 10, 32, 36, 41, 49], which has been extensively studied, but remains still an technically challenging problem. In the following, we explain the main challenges of this task to motivate the proposed new technique.

As shown in Figure 1, there are two key challenging tasks for the search-based face annotation framework: (i) how to efficiently retrieve the top-n most similar facial images from a large facial image database given a query facial image, that is, how to develop an effective content-based facial image retrieval solution; and (ii) how to effectively exploit the short list of candidate facial images and their weak labels for naming the faces automatically. In general, these two tasks can be solved separately, though the second task can be affected by the results of the first task. As one can tackle the first task by adapting existing CBIR techniques [24, 43, 7, 42], in this paper we focus on the second challenge, which is critical due to the nature of noisily labeled web facial images.

In this paper, we propose a novel framework of "*Learning to Name Faces*" (L2NF) for search-based face annotation, which attempts to learn both the *optimal weight vector* in combining different query-neighbor similarity functions for face naming and the *refined labels* for enhancing the initial weak labels simultaneously in a unified learning framework. In particular, the key challenge of naming the query facial image is to effectively measure the similarity between the query image and its nearest instances by combining diverse facial feature representations and their proper distance measurements. To tackle this challenge, we propose a multimodal learning scheme that (i) first constructs multiple diverse facial features for representing the faces, (ii) further optimizes the distance measure on each feature space (modality) using distance metric learning, and (iii) finally learns the optimal fusion of the multiple representations by adapting the structural SVM algorithm. Besides, we suggest a graph-based label refinement scheme to enhance the weak labels of top-ranked similar facial images by exploiting the "label smoothness" assumption. The main contributions of this work include:

- We propose a novel "Learning to Name Faces" (L2NF) scheme, which tackles the face naming problem by exploring multimodal learning on weakly labeled facial image data.
- We conduct extensive experiments to evaluate the proposed algorithm for face annotation on large-scale web facial image databases and obtain encouraging results.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed algorithms of Learning to Name Faces (L2NF). Section 4 shows the experimental results of our empirical studies. Section 5 discusses the limitations, and finally Section 6 concludes this paper.

2. RELATED WORK

Automated face annotation can be directly solved by general face recognition and verification techniques, which have been extensively studied for many years [20]. However, the success of such "*model-based face annotation*" scheme often relies on a large set of high-quality facial images collected in well-controlled environments, which can be difficult and expensive to obtain. This drawback has been partially addressed in recent benchmark studies of unconstrained face detection and verification techniques on facial image testbeds collected from the web, such as LFW [18, 5].

By focusing on the facial image domain, the studies for automated face annotation can be further classified into four groups. The first group of studies aim to handle the collections of personal photos [8], where rich context clues, such as social context, GPS tags, time tamps, are available. Some techniques have already been successfully deployed in the commercial applications, e.g., Apple iPhoto,[1] Google Picasa,[2] and Facebook face auto-tagging solution.[3] The second group of works consider to refine the text-based facial image retrieval results, where a human name is used as the input query [27, 22, 17, 12, 13]. Such problems are closely related to the image re-ranking problems, and part of top-ranked facial images are tagged with the name query. For example, Ozkan and Duygulu proposed a graph-based model for finding the densest sub-graph as the most related result [27], which is improved by adding an extra constraint such that a face can only appear once in an image [12] or by introducing the images of "friends" of the query name in a query expansion scheme [26]. Following the graph-based approach, Le and Satoh [22] proposed to represent the importance of each returned image. Recently, the generative approach has also been adopted in this problem and achieved better performance [12, 13]. The third group of works have attempted to directly annotate each web facial image with the names extracted from its caption information. For example, Berg et al. [3] proposed a probability model which is combined with a clustering algorithm to estimate the relationship between the facial images and the names in their captions. Guillaumin et al. [12] proposed to iteratively update the assignment between the facial image and detected names in captions based on a minimum cost matching algorithm, which is further improved by using supervised distance metric learning techniques to grasp the important discriminative features in low dimensional spaces in their subsequent work [13]. Recently, Bu et al. [4] proposed to estimate the distance between faces and names with "commute distance". The last group of studies is the "*search-based face annotation*" (SBFA), which was inspired by the search-based image annotation and is fundamentally different from the previous three groups of research. In particular, the SBFA framework aims to solve the generic content-based face annotation problem, where facial images are directly used as the input query images and the task is to return the corresponding human names for the query images. There are rather few studies in this group. For example, by attempting to mine the large-scale noisy web facial image with weak labels, Wang et al. [37] proposed an Unsupervised Label Refinement (URL) algorithm to enhance the initial weak label matrix over the entire facial image database. In their subsequent work [38], they further proposed the Weak Label Regularized Local Coordinate Coding (WLRLCC) algorithm, which aims to fully exploit the top-ranked similar images of the query facial image via a unified optimization scheme of learning both local coordinate coding and refined labels.

[1] http://www.apple.com/ilife/iphoto/

[2] http://picasa.google.com/

[3] http://www.facebook.com/

Recently, a few of emerging works have attempted to attack the automated face annotation problem via the "*search-based face annotation*" (SBFA) paradigm [37, 38]. It was generally inspired by the *search-based image annotation* that attempts to infer the correlation or joint probabilities between query images and annotation keywords based on existing object recognition techniques and semi-supervised learning algorithms in mining massive free web images on the WWW [9, 10, 6, 32, 15, 39, 29, 35, 28]. Several studies have attempted to develop efficient content-based indexing and search techniques to facilitate image tagging tasks. For example, Russell et al. [29] developed a large collection of web images with ground truth labels to facilitate object recognition tasks. More studies in this area aim to address the final annotation process by exploring effective label propagation algorithms. For example, Wright et al. [40] proposed a classification algorithm based sparse representation technique, which predicts the label information based on the class-based feature reconstruction. Tang et al. [32] presented a sparse graph-based semi-supervised learning (SGSSL) approach to annotate web images. Wang et al. [36] proposed another sparse coding based annotation framework, where the label-based graph is used to learn a linear transformation matrix for feature dimension reduction, and sparse reconstruction is employed for the subsequent label propagation step. Wu et al. [41] proposed to select heterogeneous features with structural grouping sparsity and suggested a Multi-label Boosting scheme (denoted as "MtBGS" for short) for feature regression, where a group sparse coefficient vector is obtained for each class (category) and further used for predicting new instances. Wu et al. [43] proposed a multi-reference re-ranking scheme (denoted as "MRR" for short) for improving the retrieval process.

Our work differs from the above existing works for search-based face annotation in several aspects. First of all, the ULR algorithm aims to refine the noisy labels over the entire facial image database, which is extremely time-consuming for the large-scale database. Unlike the ULR algorithm, our work tackles such a computationally expensive task by mining only the top-ranked similar images for each query, which follows the similar approach of the WLRLCC algorithm. Further, both ULR and WLRLCC algorithms are designed to explore only one single type of facial feature descriptor, e.g., the GIST features, while our work is designed to explore more clues by constructing multiple types of facial features descriptors and further learning to optimize the fusion of the multimodal representations.

3. L2NF — LEARNING TO NAME FACES

In this section, we present the proposed "Learning to Name Faces" (L2NF) framework for search-based face annotation in detail.

3.1 Preliminaries

Throughout the paper, we denote matrixes or sets by upper case letters, e.g., X and D; we denote vectors by bold lower case letters, e.g., $\mathbf{x}$, $\mathbf{x}_i$, $\mathbf{x}_{ij}$; we denote scalars by normal letters, e.g., x_i, X_{ij}, where x_i is the i-th element of vector $\mathbf{x}$ which could also be denoted as $\mathbf{x}[i]$, and X_{ij} is the element in the i-row and j-column of matrix X, which could also be denoted as $X[i,j]$.

Let us denote by $Q = \{\mathbf{q}_i | i = 1, 2, \ldots, N_t\}$ the set of query facial images, and assume there are a total of m names (classes) in the whole retrieval database, denoted by $C = \{c_1, c_2, \ldots, c_m\}$. Each query facial image $\mathbf{q}_i \in Q$ is associated with one name (class label) $c_{\mathbf{q}_i} \in C$. Notice that we assume that there is only one name for each person. We denote by $\mathbf{y}_{\mathbf{q}_i} \in \{0,1\}^m$ the label vector for the query instance $\mathbf{q}_i$, which contains only one non-zero item: $\|\mathbf{y}_{\mathbf{q}_i}\|_0 = 1$. If the query instance $\mathbf{q}_i$ is annotated with the k-th name ($c_{\mathbf{q}_i} = c_k$), then $\mathbf{y}_{\mathbf{q}_i}[k] = 1$. In the SBFA framework, the name (label) of the query face is predicted based on its nearest facial images. Assume the top-n retrieval results of the query image $\mathbf{q}_i$ are $\{(\mathbf{d}_{ij}, \mathbf{y}_{ij}) | j = 1, 2, ..., n\}$, where $\mathbf{d}_{ij}$ is the j-th similar image in the retrieval result and $\mathbf{y}_{ij} \in \{0,1\}^m$ is its corresponding label vector. We denote by $Y_i = [\mathbf{y}_{i1}, \mathbf{y}_{i2}, \ldots, \mathbf{y}_{in}] \in \mathbb{R}^{m \times n}$ the label matrix for the i-th query $\mathbf{q}_i$.

For each query-neighbor pair $(\mathbf{q}_i, \mathbf{d}_{ij})$, we can create one query-neighbor similarity based feature vector:

$$\mathbf{x}_{ij} = \mathbf{\Phi}(\mathbf{q}_i, \mathbf{d}_{ij}) = [\phi_k(\mathbf{q}_i, \mathbf{d}_{ij})]_{k=1}^{N_f}$$

where $\phi_k(\cdot,\cdot)$ represents the k-th query-neighbor similarity function and N_f is the number of the query-neighbor similarity functions. Typically, the query-neighbor similarity function is related to three factors: (1) the facial feature representation, (2) the distance metric, and (3) the mapping function between the distance value and the similarity value. For example, we can extract the Local binary patterns (LBP) as the facial feature, apply the L2-norm (Euclidean) distance as the distance metric, and the radial basis function with $\gamma = 0.1$ as the similarity-mapping function:

$$\exp(-\frac{1}{\gamma^2}\|\mathbf{q}_i^{(\mathrm{lbp})} - \mathbf{d}_{ij}^{(\mathrm{lbp})}\|^2).$$

To estimate the similarity more accurately by exploring more information, we can leverage multiple diverse query-neighbor similarity functions. More details about query-neighbor similarity function construction will be presented in Section 3.4. Based on the predefined query-neighbor similarity function and the achieved query-neighbor similarity based feature vector, for the i-th query instance $\mathbf{q}_i$, we denote its query-neighbor similarity matrix by $X_i = [\mathbf{x}_{ik}]$ with $k = 1, 2, \ldots, n$ and $X_i \in \mathbb{R}^{N_f \times n}$.

3.2 Problem Overview

The basic idea of the SBFA paradigm is to exploit the weak labels of top-ranked similar facial images for naming the query face. The crux for the face naming task lies in how to effectively estimate the confidence values for the weak label vectors of the top-ranked similar instances. Given a query image $\mathbf{q}_i$ and its top-n retrieval results $\{(\mathbf{d}_{ij}, \mathbf{y}_{ij}) | j = 1, 2, \ldots, n\}$, we denote by v_{ij} the confidence value for its j-th similar image $\mathbf{d}_{ij}$. Then, the estimated label vector of $\mathbf{q}_i$, denoted as $\hat{\mathbf{y}}_{\mathbf{q}_i}$, can be generated as:

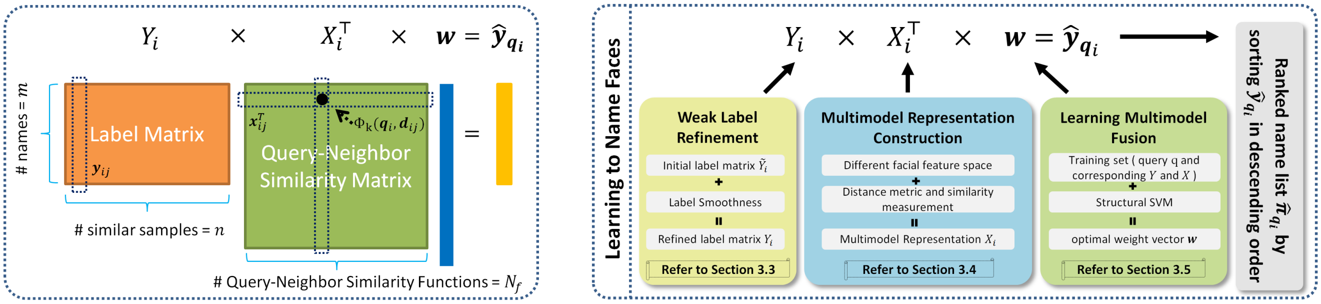
$$\hat{\mathbf{y}}_{\mathbf{q}_i} = \sum_j \mathbf{y}_{ij} * v_{ij} = Y_i \mathbf{v}_i \qquad (1)$$

where $\mathbf{v}_i = [v_{i1}, v_{i2}, \ldots, v_{in}]^\top$. Obviously, the confidence value v_{ij} is related to both query $\mathbf{q}_i$ and the j-th similar instance $\mathbf{d}_{ij}$. In our problem, we assume it linearly depends on the query-neighbor similarity based feature vector $\mathbf{x}_{ij}$, that is, $v_{ij} = \mathbf{x}_{ij}^\top \mathbf{w}$. Hence, the confidence vector $\mathbf{v}_i$ can be achieved as follows:

$$\mathbf{v}_i = X_i^\top \mathbf{w} \qquad (2)$$

where $\mathbf{w} \in \mathbb{R}^{N_f}$ is a weight vector for multimodal fusion, which aims to combine different features of X generated by the N_f diverse similarity functions. In other words, each confidence value v_{ij} is a weighted linear combination of the corresponding query-neighbor similarity based feature vector $\mathbf{x}_{ij}$.

Remark. This aforementioned assumption is not difficult to understand as follows: each item in $\mathbf{x}_{ij}$ (e.g. the k-th item $\mathbf{x}_{ij}[k]$) is only related to the corresponding similarity function (e.g. $\phi_k(\cdot,\cdot)$). A large $\mathbf{x}_{ij}[k]$ indicates that the j-th retrieved instance is more similar to the query instance based on the k-th similarity function. Hence, it is more possible that the query instance has the same label


Y_i × $X_i^\top$ × $w = \hat{y}_{q_i}$
# names = m
Label Matrix
y_{ij}
x_{ij}^T
$\Phi_k(q_i, d_{ij})$
Query-Neighbor Similarity Matrix
=
# similar samples = n
# Query-Neighbor Similarity Functions = N_f
Learning to Name Faces
Weak Label Refinement
Initial label matrix $\tilde{Y}_i$
+
Label Smoothness
=
Refined label matrix Y_i
Refer to Section 3.3
Multimodel Representation Construction
Different facial feature space
+
Distance metric and similarity measurement
=
Multimodel Representation X_i
Refer to Section 3.4
Learning Multimodel Fusion
Training set (query q and corresponding Y and X)
+
Structural SVM
=
optimal weight vector w
Refer to Section 3.5
Ranked name list $\hat{\pi}_{q_i}$ by sorting $\hat{y}_{q_i}$ in descending order

(a) (b)

Figure 2: Introduction of face naming in "*Search-based Face Annotation*". (a) A visual explanation of Eq.(3). For a query instance $\mathbf{q}_i$, its top-n similar samples are $\{(\mathbf{d}_{ij}, \mathbf{y}_{ij})\}_{j=1,2,\dots,n}$, and the predicted label vector is $\hat{\mathbf{y}}_{\mathbf{q}_i}$. $\mathbf{y}_{ij}$ is the label vector of the j-th nearest sample $\mathbf{d}_{ij}$. $\mathbf{x}_{ij}$ is the feature vector between the query instance $\mathbf{q}_i$ and the similar example $\mathbf{d}_{ij}$. The k-th item of $\mathbf{x}_{ij}$ is constructed with the k-th query-neighbor similarity function $\phi_k(\mathbf{q}_i, \mathbf{d}_{ij})$. The inter product value $\mathbf{x}_{ij}^\top \mathbf{w}$ is the confidence values v_{ij} for the j-th label vector $\mathbf{y}_{ij}$. (b) Three factors that affect the annotation performance of SBFA and the corresponding improvement solution.

vector with the j-th retrieved instance. As a result, the similarity value $\mathbf{x}_{ij}[k]$ is correlated with the the confidence value v_{ij}. Typically, different similarity functions can perform very differently in practice, hence they should be combined appropriately by a proper weight vector $\mathbf{w}$

By combining Eq.(1) and Eq.(2), the estimated label vector $\hat{\mathbf{y}}_{\mathbf{q}_i}$ for the query image $\mathbf{q}_i$ can be computed as follows:

$$\hat{\mathbf{y}}_{\mathbf{q}_i} = Y_i \mathbf{v}_i = Y_i X_i^\top \mathbf{w} \tag{3}$$

where Y_i and X_i vary for different query instances, and $\mathbf{w}$ is independent of the query instances. We show a visual example in Figure 2 (a) to help understand this formula.

To generate the final annotation results (a sorted candidate name list), we can rank all the m names by sorting the predicted label vector $\hat{\mathbf{y}}_{\mathbf{q}_i}$ in a descending order, as shown in Figure 2 (b). We denote by $\hat{\boldsymbol{\pi}}_{\mathbf{q}_i}$ the ranked name list, in which the item $\hat{\boldsymbol{\pi}}_{\mathbf{q}_i}[j] \in C$ is the j-th annotated name. Given the correct name of the query instance $\mathbf{q}_i$ is $c_{\mathbf{q}_i}$, a good annotation system should ensure $c_{\mathbf{q}_i}$ appears at the top-ranked position or ideally at the first position. Hence, our problem aims to minimize the ranking position of the correct name $c_{\mathbf{q}_i}$, which can be formulated as follows:

$$\min_{\mathbf{w}, Y_i, X_i} \sum_{i=1}^{N_t} \text{loss}(c_{\mathbf{q}_i}, \hat{\boldsymbol{\pi}}_{\mathbf{q}_i}) \tag{4}$$

$$\text{where } \hat{\boldsymbol{\pi}}_{\mathbf{q}_i} = \text{rank}(\hat{\mathbf{y}}_{\mathbf{q}_i}) \text{ and } \hat{\mathbf{y}}_{\mathbf{q}_i} = Y_i X_i^\top \mathbf{w}$$

In general, the loss value should be zero if the correct name $c_{\mathbf{q}_i}$ is at the first position of $\hat{\boldsymbol{\pi}}_{\mathbf{q}_i}$, and the loss value of $c_{\mathbf{q}_i}$ at the top-ranked position should be smaller than the one of $c_{\mathbf{q}_i}$ at a lower-ranked position. The goal of the whole learning to name faces scheme is to minimize the loss values over all the query instances by addressing the following three key factors: (i) the noisy label matrix Y_i, (ii) the query-neighbor similarity matrix X_i, and (iii) the combination weight vector $\mathbf{w}$, as shown in Figure 2 (b). In particular, we attempt to address each of them respectively in the following approach:

- To address the noisy nature of web images, we propose to refine the initial weak label information Y_i by a graph-based refinement scheme for exploiting the "label smoothness" assumption;
- To address the variances of web facial images captured under various conditions (illumination, position, age, and gender, etc.), we can construct multiple diverse query-neighbor similarity functions and further improve the similarity measurements by employing distance metric learning techniques;
- To find the optimal multimodal fusion, we propose a supervised learning to rank scheme to optimize the weight vector $\mathbf{w}$ by applying the structural SVM algorithms on a set of training query samples (N_t query images and their corresponding top-n retrieval results).

In the following, we will introduce the solutions of the aforementioned three problems respectively.

3.3 Weak Label Refinement

In this section, we aim to refine the initial weak label matrix for each query independently. In particular, for a query $\mathbf{q}$ (the subscript of query index value is omitted), its top-n similar samples are $\{\mathbf{d}_1, \dots, \mathbf{d}_n\}$ and the corresponding noisy label matrix is denoted by $\tilde{Y}$. We enhance the initial label matrix $\tilde{Y}$ in a manifold learning scheme based on the key assumption of "*label smoothness*", which means that the more similar the visual contents of two facial images are, the more likely they share the same labels [37].

In particular, for two images $\mathbf{d}_i$ and $\mathbf{d}_j$ in the top-n nearest samples, we can compute their similarity value vector based on the query-neighbor similarity function: $\mathbf{\Phi}(\mathbf{d}_i, \mathbf{d}_j) \in \mathbb{R}^{N_f}$. By using the weight vector $\mathbf{w}$ learned in Section 3.5, we can get the similarity value between $\mathbf{d}_i$ and $\mathbf{d}_j$ as $S_{ij} = \mathbf{w}^\top \mathbf{\Phi}(\mathbf{d}_i, \mathbf{d}_j)$. A large value of S_{ij} indicates that $\mathbf{d}_i$ is more similar to $\mathbf{d}_j$. Hence, a larger value of S_{ij} implies that the label vectors of $\mathbf{d}_i$ and $\mathbf{d}_j$ are more likely to be the same. Based on the above motivation, we can obtain the following formulation to enhance the initial weak label matrix $\tilde{Y}$:

$$\min_{Y \geq 0} \sum_{i,j} S_{ij} * \|Y_{:i} - Y_{:j}\|^2 + \beta \|(Y - \tilde{Y}) \circ M\|_F^2 \tag{5}$$

where $\circ$ denotes the Hadamard product of two matrices, and M is the mask matrix indicating the non-zeros values in $\tilde{Y}$. In Eq.(5), the first term enforces the "*label smoothness*" assumption. Following the previous work [38], the second term is a regularization term that prevents the refined label matrix being deviated too much from the initial weak matrix $\tilde{Y}$. Notice that the label refinement problem in Eq.(5) depends on the weight vector $\mathbf{w}$ achieved by Eq.(9), while learning the weight vector $\mathbf{w}$ depends on the input data Y, as shown in Eq.(9). In our problem, we update the label matrices $Y_i, i = 1, \dots, n$ and the weight vector $\mathbf{w}$ iteratively.

3.4 Multimodal Representation Construction

In this section, we aim to construct the multimodal representation between the query instances and their corresponding top-ranked similar samples, which is based on the query-neighbor similarity function: $\mathbf{\Phi} = \{\phi_k\}_{k=1,2,\ldots,N_f}$. Generally, we can represent one facial image in different feature space, e.g. LBP feature, GIST feature, and Gabor feature. Suppose there are $\mathcal{K}$ kinds of features in total, we can represent by $(\mathbf{q}^{(k)}, \mathbf{d}_i^{(k)})$ the query-neighbor feature pair between the query image $\mathbf{q}$ and its i-th nearest sample $\mathbf{d}_i$ in the k-th feature space. Following the existing works on distance metric learning [47, 46], we can define a distance metric $M^{(k)}$ in the k-th feature space, hence, the distance between $\mathbf{q}^{(k)}$ and $\mathbf{d}_i^{(k)}$ can be expressed by

$$d_{M^{(k)}}(\mathbf{q}^{(k)}, \mathbf{d}_i^{(k)}) = \sqrt{(\mathbf{q}^{(k)} - \mathbf{d}_i^{(k)})^\top M^{(k)} (\mathbf{q}^{(k)} - \mathbf{d}_i^{(k)})}$$

and the inner product between $\mathbf{q}^{(k)}$ and $\mathbf{d}^{(k)}$ can be expressed by

$$< \mathbf{q}^{(k)}, \mathbf{d}_i^{(k)} >_{M^{(k)}} = (\mathbf{q}^{(k)})^\top M^{(k)} (\mathbf{d}_i^{(k)})$$

Based on the k-th feature space and distance matrix $M^{(k)}$, there are two ways to compute the similarity values between two instances: one way is using the heat kernel to transform the distance value into a similarity value which is widely used in semi-supervised learning [2]. In detail, the similarity value between $\mathbf{q}^{(k)}$ and $\mathbf{d}_i^{(k)}$ can be computed as follows:

$$\phi(\mathbf{q}, \mathbf{d}_i; k, \gamma) = \exp(-d^2_{M^{(k)}}(\mathbf{q}^{(k)}, \mathbf{d}_i^{(k)})/\gamma^2), \quad \gamma \in \Gamma$$

where the query-neighbor similarity function ϕ depends on the feature type k and the parameter γ. As a result, we can obtain $k * |\Gamma|$ query-neighbor similarity functions, where Γ is the set of all possible parameters γ during the experiments.

Another way to compute the similarity value is using the sparse representation technique which has been adopted to construct adjacency matrix in some recent works [32]. In detail, in the k-th feature space with $M^{(k)}$ as the distance metric, we can obtain the sparse representation $\mathbf{s}^{(k)}$ for $\mathbf{q}^{(k)}$ based on the dictionary $D^{(k)} = [\mathbf{d}_1^{(k)}, \ldots, \mathbf{d}_n^{(k)}]$ with the kernelized sparse coding algorithm [11], which can be formulated as follows:

$$\begin{aligned} \mathbf{s}_\lambda^{(k)} = \arg \min_{\mathbf{s}^{(k)} \geq 0} < \mathbf{q}^{(k)}, \mathbf{q}^{(k)} >_{M^{(k)}} &+ (\mathbf{s}^{(k)})^\top K_{DD}^{(k)} (\mathbf{s}^{(k)}) \\ &- 2(\mathbf{s}^{(k)})^\top K_{D\mathbf{q}}^{(k)} + \lambda \|\mathbf{s}^{(k)}\|_1 \end{aligned}$$

where $\mathbf{s}_\lambda^{(k)}$ is the achieved sparse representation with parameter λ, $K_{DD}^{(k)}$ is an $n \times n$ matrix with $\{K_{DD}^{(k)}\}_{ij} =< \mathbf{d}_i^{(k)}, \mathbf{d}_j^{(k)} >_{M^{(k)}}$, and $K_{D\mathbf{q}}^{(k)}$ is an $n \times 1$ vector with $\{K_{D\mathbf{q}}^{(k)}\}_i =< \mathbf{q}^{(k)}, \mathbf{d}_i^{(k)} >_{M^{(k)}}$. The i-th item in the sparse represent $\mathbf{s}_\lambda^{(k)}$ presents the representative ability of the i-th dictionary instance $\mathbf{d}_i^{(k)}$ for the encoding instance $\mathbf{q}^{(k)}$. Hence, the similarity value between $\mathbf{q}^{(k)}$ and $\mathbf{d}_i^{(k)}$ is computed as follows:

$$\phi(\mathbf{q}, \mathbf{d}_i; k, \lambda) = \mathbf{s}_\lambda^{(k)}[i], \quad \lambda \in \Lambda$$

where the query-neighbor similarity function ϕ depends on the feature type k and the parameter λ. As a result, we can obtain $k * |\Lambda|$ query-neighbor similarity functions, where Λ is the set of all possible parameters Λ during the experiments.

Finally, for each feature space, we must choose a distance metric $M^{(k)}$. We can use the original feature space by setting $M^{(k)}$ with the identify matrix. To keep all the data points within the same classes close and separate all the data points from different classes far apart, it is better to adopt distance metric learning (DML) techniques to learn a better distance metric for each feature space respectively. Generally, any supervised DML algorithms can be used since the query-neighbor similarity function ϕ is independent of the DML algorithms. In our problem, we adopt “Metric Learning to Rank" (MLR) algorithm [25] that learns a metric such that rankings of data induced by the learned distance are optimized against a ranking loss measure (e.g. ROC area (AUC) or MAP). In this setting, the “relevant" results (in the same class) should lie close in space to the query, and “irrelevant" results should be pushed far away.

3.5 Optimal Fusion of Multiple Modalities

In this section, we aim to find the optimal weight vector $\mathbf{w}$ for optimizing the multimodal fusion. In particular, given a label matrix Y_i and a query-neighbor similarity matrix X_i, we can directly achieve the annotation result of $\mathbf{q}_i$ with the fusion vector $\mathbf{w}$ according to Eq.(3). Hence, finding the optimal fusion vector $\mathbf{w}$ that achieves the best ranked name list in Eq.(4) is equivalent to learning a multimodal annotation function with parameter $\mathbf{w}$ as follows:

$$f(\mathbf{w}) : \mathcal{Y} \times \mathcal{X} \to \Pi$$

based on a set of training samples $\{(\mathbf{q}_i, \mathbf{y}_{\mathbf{q}_i}, L_i, X_i)\}$ with $i = 1, 2, \ldots, N_t$ by minimizing the annotation errors. The input space contains all the multiplication results between label matrix Y_i and query-neighbor similarity matrix X_i. The output space Π contains all the possible annotation results (the ranked name list). Obviously, the result of the function f is a structural output instead of a scalar value. Hence, it could be formulated as a structural SVM problem [34, 19], which has been extensively studied in several research works and has been used for ranking problems in [48, 25, 45]. To specialize a general structure SVM algorithm for a particular problem, we define two functions: the “*loss function*" Δ and the “*feature combination function*" Ψ.

3.5.1 Loss Function

The loss function is denoted as $\Delta(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}})$ in our problem, where $\boldsymbol{\pi}$ is the ground-truth ranked name list generated by the ground-truth label vector $\mathbf{y}$, while $\hat{\boldsymbol{\pi}}$ is the predicted ranked name list generated by the predicted label vector $\hat{\mathbf{y}}$. Notice that we omit the subscript for the query index for clarity. The “*hit rate*" at the top t annotated names is used as the performance metric, which measures the likelihood of having the correct name among the top t annotated names. In real world applications, we prefer a high hit rate value with a small t value, that is, the correct names are at the top-ranked position in the ranked name list $\hat{\boldsymbol{\pi}}$.

For one query facial image, suppose it has t_1 correct names ($t_1 = 1$ in our problem since we assume there is only one name for each person), all the correct names are at the top positions of the ground-truth name list $\boldsymbol{\pi}$, followed by all the incorrect names. For the predicted name list $\hat{\boldsymbol{\pi}}$, if we only consider its top t_2 names, the loss function can be formulated as follows:

$$\Delta(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}}) = 1 - \sum_{i=1}^{t_1} \sum_{j=1}^{t_2} h_1(\pi_i, \hat{\pi}_j) * \frac{1}{j} \tag{6}$$

where $h_1(\cdot, \cdot)$ is a judgement function that equals 1 if the i-th name π_i in $\boldsymbol{\pi}$ is the same with the j-th name $\hat{\pi}_j$ in $\hat{\boldsymbol{\pi}}$, and 0 otherwise. For example, if $t_2 = 1$, we focus on only the first annotated name which means that if the first name in $\hat{\boldsymbol{\pi}}$ is correct, then the loss value is 0, otherwise, the loss value is 1. If $t_2 = m$, the loss function in Eq.(6) becomes a special case of MAP loss.

3.5.2 *Structural-based Feature Combination*

Typically, the feature combination function aims to combine a set of feature vectors based on a ranking result. In our problem, the ranking result is denoted by $\boldsymbol{\pi}$, which contains all the m names in the name set $C = \{c_1, \ldots, c_m\}$. For a query facial image $\mathbf{q}$, its name vector is $\mathbf{y} \in \{0, 1\}^m$ with $\|\mathbf{y}\|_0 = 1$ since each facial image has only one correct name. Hence, $y_k = 1$ indicates that the k-th name c_k in C is the correct name for the query image $\mathbf{q}$. We denote by I_1 the index set of all the correct names, which contains only one item (k) according the previous case. Similarly, we denote by I_2 the index set of all the incorrect names, which contains $m-1$ items $\{1, \ldots, k-1, k+1, \ldots, m\}$. Following the previous work [48], we define the feature combination function Ψ to combine the input label matrix Y and the input query-neighbor similarity matrix X based on a ranked name list $\boldsymbol{\pi}$, as shown in Eq.(7):

$$\Psi(Y, X, \boldsymbol{\pi}) = \frac{1}{|I_1| * |I_2|} \sum_{i \in I_1} \sum_{j \in I_2} h_2(c_i, c_j, \boldsymbol{\pi})[XY_{i:}^\top - XY_{j:}^\top] \tag{7}$$

where $Y_{k:}$ is the k-th row of the label matrix L, and $h_2(\cdot, \cdot, \cdot)$ is a ranking judgement function. If the name c_i is ranked before c_j in the ranked name list $\boldsymbol{\pi}$, $h_2(c_i, c_j, \boldsymbol{\pi}) = 1$; otherwise, $h_2(c_i, c_j, \boldsymbol{\pi}) = -1$ if c_j is ranked before c_i.

Remark. For one group of input data (Y, X,$\mathbf{w}$), we can compute the label vector with Eq.(3): $\mathbf{y} = YX^\top\mathbf{w}$, which is used to generate the ranked name list $\boldsymbol{\pi}$ subsequently following the previous discussion. As shown in [45], based on the feature combination function in Eq.(7), we can obtain the same ranked name list by solving the following problem:

$$\tilde{\boldsymbol{\pi}} = \arg\max_{\boldsymbol{\pi} \in \Pi} F(\mathbf{w}, Y, X, \boldsymbol{\pi}) = \mathbf{w}^\top \Psi(Y, X, \boldsymbol{\pi}) \tag{8}$$

where $F(\mathbf{w}, Y, X, \boldsymbol{\pi})$ is the discriminant function. It indicates that we can learn the weight vector $\mathbf{w}$ by maximizing the discriminant function $F(\mathbf{w}, L, X, \boldsymbol{\pi})$ over the a set of correct ranked label lists, and predict the new label vector of the unseen query with Eq.(3).

Using the *loss function* in Eq.(6) and the *feature combination function* in Eq.(7), we can obtain the objective function to learn the weight vector $\mathbf{w}$ based on the structural SVM, which is shown as follows:

$$\min_{\mathbf{w}, \Xi = [\xi_1, \ldots, \xi_{N_t}]} \tfrac{1}{2}\mathbf{w}^\top\mathbf{w} + \tfrac{C}{N_t}\textstyle\sum_{i=1}^{N_t} \xi_i \tag{9}$$

$$s.t. \quad \forall i, \xi_i \geq 0 \quad \text{and} \quad \forall i, \forall \boldsymbol{\pi}_{\mathbf{q}_i} \in \Pi_i^\star, \forall \boldsymbol{\pi} \in \Pi \backslash \Pi_i^\star :$$

$$\mathbf{w}^\top \Psi(Y_i, X_i, \boldsymbol{\pi}_{\mathbf{q}_i}) - \mathbf{w}^\top \Psi(Y_i, X_i, \boldsymbol{\pi}) \geq \Delta(\boldsymbol{\pi}_{\mathbf{q}_i}, \boldsymbol{\pi}) - \xi_i$$

Algorithm 1: Cutting plane algorithm for Eq.(9)

```
Input: $(\mathbf{q}_i, \mathbf{y}_{\mathbf{q}_i}, X_i, Y_i), i = 1, 2, \ldots, N_t, C, \epsilon$
Output: weight vector $\mathbf{w}$
$\mathcal{W}_i \leftarrow \emptyset$, for all $i = 1, \ldots, N_t$
repeat
    for $i = 1$ to $N_t$ do
        $E(\boldsymbol{\pi}; \mathbf{w}) \equiv \Delta(\boldsymbol{\pi}_{\mathbf{q}_i}, \boldsymbol{\pi}) + \mathbf{w}^\top \Psi(Y_i, X_i, \boldsymbol{\pi})$
        $\tilde{\boldsymbol{\pi}} = \arg\max_{\boldsymbol{\pi} \in \Pi} E(\boldsymbol{\pi}; \mathbf{w})$
        if $E(\tilde{\boldsymbol{\pi}}, \mathbf{w}) > \xi_i + \epsilon$ then
            $\mathcal{W}_i \leftarrow \mathcal{W}_i \cup \{\tilde{\boldsymbol{\pi}}\}$
            Get $(\mathbf{w}, \Xi)$ by solve Eq.(9) over $\mathcal{W} = \bigcup_i \mathcal{W}_i$
        end
    end
until no $\mathcal{W}_i$ has changed during iteration;
```

In the above formulation, the objective function of Eq.(9) is similar to that of the general SVM algorithm, where C is a regularization parameter to tradeoff between the training error and the model complexity. For the constraints, if the value of discriminant function F in Eq.(8) for an incorrect ranking $\boldsymbol{\pi} \in \Pi \backslash \Pi_i^\star$ is greater than that for one true ranking $\boldsymbol{\pi}_{\mathbf{q}_i} \in \Pi_i^\star$, the slack variable ξ_i must be at least $\Delta(\boldsymbol{\pi}_{\mathbf{q}_i}, \boldsymbol{\pi})$, which indicates the sum of slacks $\sum_i \xi_i$ upper bounds the empirical risk for the training samples based on the loss function defined in Eq.(6). Since the number of constraints in Eq.(9) is extremely large, we adopt the cutting plane algorithm [19, 48] to efficiently solve the optimization in Eq.(9), as shown in Algorithm 1. More details about the cutting plane algorithm can be found in [19].

3.6 Algorithm for Learning to Name Faces

In the above, we separately discuss the three key factors that affect the final annotation result of the proposed SBFA framework, including the label matrix Y, the query-neighbor similarity matrix X and the weight vector $\mathbf{w}$, which collectively determine the annotation result as $\mathbf{y} = YX^\top\mathbf{w}$. In this section, we will present the overall training for unifying all these three factors, and how to apply the models learned by L2NF for on-the-fly face annotation of a novel query facial image.

Algorithm 2: L2NF—Algorithm for training the models

```
Input: Training set $(\mathbf{q}_i, \mathbf{y}_{\mathbf{q}_i}, \mathbf{d}_{ij}, \mathbf{y}_{ij})$ in $\mathcal{K}$ feature spaces
       with $i = 1, \ldots, N_t$ and $j = 1, \ldots, n$, name sets $C$
       with $m$ names, parameters $\beta$, $\Lambda$ and $\Gamma$
Output: weight vector $\mathbf{w}$ and similarity function set $\mathbf{\Phi}$
for $k = 1$ to $\mathcal{K}$ do
    Learn the optimal distance metric $M^{(k)}$ in Section 3.4
end
Build query-neighbor similarity functions $\mathbf{\Phi}$ with varied
combinations of $\lambda \in \Lambda$, $\gamma \in \Gamma$, and $M^{(k)}$, $k = 1, 2, \ldots, \mathcal{K}$
Construct query-neighbor feature matrix $X_i$, $i = 1, \ldots, N_t$
repeat
    Get the weight vector $\mathbf{w}$ by solving Eq.(9)
    for $i = 1$ to $N_t$ do
        Refine the label matrix $Y_i$ by solving Eq.(5)
    end
until CONVERGENCE;
```

Algorithm 2 shows the overall algorithmic framework for training the models by L2NF. At the beginning, we attempt to optimize each of the distance metrics for each facial feature space using the distance metric learning technique as discussed in Section 3.4. After obtaining the set of optimal distance metrics $M^{(k)}$, we can then construct the set of query-neighbor similarity functions $\mathbf{\Phi}$ based on the set of multiple diverse facial feature representations and their distance measures. Using the query-neighbor similarity functions $\mathbf{\Phi}$, we can generate the query-neighbor feature matrices X_i for each query q_i in the training query set. Finally, we optimize both the optimal weight vector $\mathbf{w}$ and the refined label matrices Y by an iterative scheme. At the end of the whole training scheme, we obtain the final model that consists of the set of query-neighbor similarity functions $\mathbf{\Phi}$ and the optimal weight vector $\mathbf{w}$ for multimodal fusion.

The above training framework as shown in Algorithm 2 can be done in an off-line learning manner. After completing the training, we can apply the model for online face annotation for naming a novel query facial image on-the-fly. Algorithm 3 summarizes the proposed algorithm for on-the-fly annotation of an unseen query facial image. Specifically, given a new query facial image, we first find a short list of most similar faces based on CBIR techniques.

```
Algorithm 3: Algorithm of on-the-fly face annotation by L2NF
Input: Novel query $\mathbf{q}$ in $\mathcal{K}$ feature spaces, query-neighbor
       similarity function $\mathbf{\Phi}$, optimal weight vector $\mathbf{w}*$
Output: Annotation result $\boldsymbol{\pi}$
1 Retrieval the top-$n$ similar images the $\{(\mathbf{d}_i, \mathbf{y}_i)\}_{i=1,2,\ldots,n}$
2 Construct the query-neighbor similarity matrix $X$ based on the
  query-neighbor similarity function $\mathbf{\Phi}$
3 Obtain Y by refining the initial label matrix with Eq.(5)
4 $\mathbf{y} = YX^\top \mathbf{w}*$
5 Get the ranked names list $\boldsymbol{\pi}$ by sorting $\mathbf{y}$ in descending order
```

After that, we construct the query-neighbor similarity matrix X based on the query-neighbor similarity function $\mathbf{\Phi}$. We then refine the initial label matrix Y of the current query using Eq.(5). Finally, we compute the label vector $\mathbf{y}$ by Eq.(3) and obtain the final annotation result $\boldsymbol{\pi}$ by sorting the label vector $\mathbf{y}$ in descending order.

4. EXPERIMENTAL RESULTS

4.1 Experimental Testbed

Some web facial image databases are available on the WWW, which are used in some previous research works, e.g, LFW [18],[4] Label Yahoo!News [14],[5] and FAN-Large.[6] Although the number of persons in these three databases is large, the number of images for each person is quite small. For example, there are $13,233$ images of $5,749$ people in the LFW database. The recent database PubFig [21][7] is different from these databases. In detail, it was constructed by collecting online news sources. It contains 200 persons and $58,797$ images. Due to the copyright issue, only image URL addresses are released. As some URL links are not available any more, 41,609 images are collected by our crawler in total. For each downloaded image, we crop the face image out according the provided face position rectangle and resize all the face images into the same size (128 × 128). We construct the query set by randomly collecting 10 images per person from the whole PubFig database. Hence, there are a total of $2,000$ test query images used for performance evaluation, while the rest $39,609$ images are used as the retrieval database. To construct the training set, we randomly collect $2,000$ images in the same way from the retrieval database, with the rest $37,609$ images as the retrieval database for the training set. Several facial images samples are shown in the first row of Figure 3.

Figure 3: Face image examples in Pubfig database (the first row) and WDB database (the second row)

To evaluate the L2NF framework on weakly labeled web facial images, we use another western celebrity database: "weakly labeled web facial image database" (WDB for short), which has been released in [38]. There are a total of $1,600$ query images with ground truth in the WDB database. In our experiment, we divide these queries into two parts of equal size, and randomly choose one part for model training. In "WDB" database, there are a total of four retrieval databases of different sizes. In our experiment, we use two sub-databases of different scales: "WDB-040K" and "WDB-600K". "WDB-040K" is a smaller database with $53,448$ images belonging to 400 persons, while "WDB-600K" is a large-scale database with $714,454$ belonging to $6,000$ persons. All the facial images were aligned into the same well-defined positions by the face alignment algorithms in [44], as shown in the second row of Figure 3.

To construct the query-neighbor similarity functions, we adopt three kinds of features as the facial descriptors: the LBP feature [1], the GIST feature [31, 38], and the Gabor feature [23]. In particular, the 2891-dimensional LBP feature is extracted by dividing the face images into 7×7 blocks. To reduce the computation complexity, the LBP feature is further projected into a lower 500-dimensional feature space using Principal Component Analysis (PCA). Both GIST features and Gabor features are extracted over the whole aligned facial images. The parameter set for heat kernel is $\Gamma = \{-1, 0, 1, 2, 3, 4\}$, while the parameter set for sparse representation is $\Lambda = \{0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$. The parameters β and C of L2NF are set as 10 and 1000, respectively. For the distance metric learning algorithm (MLR), the parameter C is set as 10.

Following the previous works, we adopt the *hit rate* at top-T annotated results as the performance metric to evaluate the annotation performance, which measures the likelihood of having the true label among the top-T annotated names for a query facial image. We compare the proposed "L2NF" framework with several existing works that are proposed for web-scale face annotation or general image annotation, including "WLRLCC" [38], "SGSSL" [32], "MtBGS" [41], "MRR" [43], and a simple baseline algorithm that simply adopts the weighted majority voting "WMV". We also extend the WLRLCC algorithm and the WMV algorithm into a multimodal scheme, by equally combining the face naming results from different facial feature spaces, denoted as "WLRLCC$_{\mathrm{mm}}$" and "WMV$_{\mathrm{mm}}$".

4.2 Experiments on "WDB-040K"

This experiment aims to evaluate the face naming performance of the proposed "L2NF" framework on the database "WDB-040K" by comparing with the aforementioned seven existing algorithms. For the facial image retrieval task in L2NF, we adopt the JEC algorithm to combine the distances from different face descriptors [24], which allows each individual distance to contribute equally. The same retrieval scheme is used to find the top-ranked similar images for the multimodal extensions: "WLRLCC$_{\mathrm{mm}}$" and "WMV$_{\mathrm{mm}}$". For the single model solution, we use the GIST feature as the facial descriptor, which is similar to the experiment setting in [38]. For the $1,600$ query facial images, we randomly select half of them to learn the distance metrics of different facial features and the multimodal representation combination $\mathbf{w}$. Such a procedure is repeated 10 times and the average performance is computed over the 10 trials, as shown in Table.1.

Several observations can be drawn from the results. First, for the the single model solution, the WLRLCC algorithms achieves the best performance by using only one type of facial feature (GIST). In detail, the simple baseline WMV is about 60.9% with $T = 1$, which is boosted to 76.7% by WLRLCC. Second, if multiple facial features are available, the performance of the multimodal WLRLCC$_{\mathrm{mm}}$ increases to 80.9%, and 65.6% for the multimodal WMV$_{\mathrm{mm}}$. It indicates that using multiple facial representations is

[4] http://goo.gl/4EuI1

[5] http://goo.gl/2XlES

[6] http://goo.gl/2baSv

[7] http://goo.gl/zlb4l

Table 1: Face naming performance on database "WDB-040K".

	T=01	T=02	T=03	T=04	T=05
WMV	0.6090	0.7150	0.7599	0.7848	0.7969
	± 0.012	± 0.009	± 0.009	± 0.008	± 0.008
SGSSL	0.7310	0.7770	0.8079	0.8231	0.8338
	± 0.011	± 0.010	± 0.011	± 0.013	± 0.012
MtBGS	0.7023	0.7654	0.7896	0.8058	0.8215
	± 0.015	± 0.012	± 0.011	± 0.009	± 0.010
MRR	0.6640	0.7560	0.7875	0.8005	0.8155
	± 0.010	± 0.007	± 0.008	± 0.008	± 0.009
WLRLCC	0.7671	0.8009	0.8263	0.8361	0.8496
	± 0.010	± 0.010	± 0.010	± 0.009	± 0.007
$\mathrm{WMV}_{\mathrm{mm}}$	0.6560	0.7491	0.8010	0.8144	0.8244
	± 0.014	± 0.013	± 0.012	± 0.013	± 0.014
$\mathrm{WLRLCC}_{\mathrm{mm}}$	0.8088	0.8568	0.8714	0.8799	0.8859
	± 0.011	± 0.014	± 0.015	± 0.012	± 0.011
L2NF	**0.8663**	**0.8918**	**0.8983**	**0.9025**	**0.9054**
	± 0.011	± 0.011	± 0.010	± 0.010	± 0.009

helpful for the face naming task which validates the importance of this study. More specifically, the improvements of these two algorithms ($\mathrm{WLRLCC}_{\mathrm{mm}}$ and $\mathrm{WMV}_{\mathrm{mm}}$) in the multimodal scheme are mainly gained from two aspects: (i) the retrieval result becomes better when multiple features and distance measures are combined by JEC [24]. For example, for the WMV algorithm, if we use the multiple features for the retrieval step but only use GIST feature for the annotation step, its performance is 64.2%, which is higher than the one that uses only GIST feature for both retrieval and annotation steps(60.9%). (ii) the combination enlarges the probability that the correct name is chosen. Both of the two aspects are beneficial for the L2NF framework. Last but not least, the proposed L2NF framework can further improve the face naming performance to 86.6%, which indicates the constructed multimodal representations are discriminative and the learned fusion vector can efficiently combine various query-neighbor similarity function in different facial feature spaces. The performance improvement is mainly gained from three aspects: the refined label matrix, the constructed multimodal representation based on distance metric learning techniques, and the learned optimal combination of various query-neighbor similarity functions. More details will be further discussed in Section 4.5.

4.3 Experiments on "WDB-600K" & "PubFig"

This experiment aims to evaluate the face naming performance of the proposed L2NF framework on two different larger facial image databases: "WDB-600K" and "PubFig". The two databases were collected under very different approaches and settings, which can help us evaluate the generalization of the proposed technique on real-world data under different scenarios. For clarity, we mainly focus on the evaluation of the algorithms using multimodal representations. The experimental results are shown in Figure 4 and Table 2.

We can make several observations from the results. First of all, similar to the previous observations, the proposed L2NF framework consistently achieves the best annotation performance among all the compared algorithms. It shows that for different databases the proposed L2NF algorithm is always helpful to improve the annotation performance. Secondly, the performance on "WDB-600K" is lower than the one on "WDB-400K" which is consistent with the previous observation in [38], since increasing the number of persons leads to a larger database of more images, which makes the retrieval task more challenging. Finally, it is interesting to observe the overall annotation performance on the PubFig database is worse than the results on WDB-series databases. There are several reasons for this observation: (i) the number of images per person varies a lot in the PubFig database, in which several persons own only about 20 images. It is insufficient for the data-driven scheme, hence the annotation performance is reduced; (ii) All the facial images in the PubFig database are cropped according to the face position without adopting any face alignment algorithms, which makes the facial descriptor sensitive for face views.

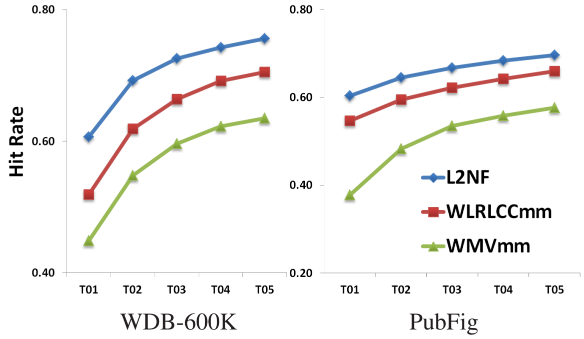


Figure 4: Face naming performance (hit rate @ top-T) on the two databases: "WDB-600K" and "PubFig".

4.4 Evaluation on Training Query Sizes

This experiment aims to evaluate the impact of the training query sizes in the L2NF framework based on the "WDB-040K" database. In the previous experiments, we adopt half of the $1,600$ query images as the training set. In this experiment, we evaluate the annotation performance under varied number of training query images. Specifically, instead of using all the training samples (totally 800), we build three small training sets by randomly collecting 200, 400 and 600 query images, respectively. The experimental results of hit-rate @ top-1 performance are shown in Figure 5. From the results, it is obvious that the face naming performance increases when more training samples are available, and the final performance tends to become saturated when the training query size is above 600. Finally, even with a small number of queries for training, e.g., only 200 training samples, the L2NF algorithm can achieve a good performance (83.4%), which remains much better than the state-of-the-art "$\mathrm{WLRLCC}_{\mathrm{mm}}$" scheme.

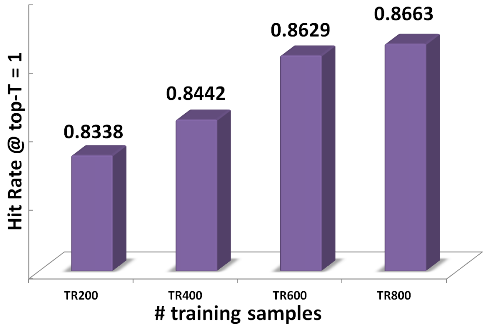


Figure 5: Face naming performance (hit rate @ top-$T = 1$) of L2NF with varied sizes of training queries.

4.5 Analysis of the Performance Gains

This experiment aims to analyze how different factors affect the face naming performance by the proposed L2NF scheme as shown in Figure 2 (b). In particular, there are three key factors: the refined

Table 2: Face naming performance (Hit Rate) on database “WDB-600K" and “PubFig"

	Database: WDB-600K					Database: PubFig				
	T=01	T=02	T=03	T=04	T=05	T=01	T=02	T=03	T=04	T=05
WLRLCC	0.5150	0.5900	0.6400	0.6675	0.6813	0.5369	0.5911	0.6213	0.6431	0.6604
WMV	0.4538	0.5500	0.5950	0.6163	0.6413	0.3572	0.4616	0.5172	0.5453	0.5650
WMV_{mm}	0.4488	0.5475	0.5963	0.6225	0.6350	0.3775	0.4830	0.5352	0.5585	0.5763
$\text{WLRLCC}_{\text{mm}}$	0.5188	0.6188	0.6638	0.6913	0.7050	0.5468	0.5948	0.6218	0.6423	0.6595
L2NF	**0.6065**	**0.6920**	**0.7258**	**0.7426**	**0.7559**	**0.6034**	**0.6449**	**0.6674**	**0.6839**	**0.6961**

label matrix, the constructed multiple representations, and the optimal weight vector for multimodal fusion.

Table 3: Evaluation and analysis of the performance gains.

	$\text{L2NF}^{\mathbf{w}=1}_{M=I}$	$\text{L2NF}^{\mathbf{w}=\mathbf{w}^\star}_{M=I}$	$\text{L2NF}^{\mathbf{w}=1}_{M=M^\star}$	$\text{L2NF}^{\mathbf{w}=\mathbf{w}^\star}_{M=M^\star}$
Hit Rate	0.7941	0.8120	0.8403	0.8663

First of all, to examine the efficacy of the refined label matrix Y, we compare it with the initial raw label matrix $\tilde{Y}$ using the simplest baseline algorithm WMV by excluding other factors in affecting annotation performance. Our result indicates that the refined label matrix can boost the performance from 60.9% (without refinement) to 62.0% (after refinement). Further, we examine the efficacy of another two factors as shown in Table 3. We denote by $M^\star$ the learned distance metric and $\mathbf{w}^\star$ the optimal multimodal fusion vector. When the weight vector $\mathbf{w}$ is fixed to 1 and the distance metrics are based on Euclidean distance (M is set to an identity matrix), the hit-rate @ top-1 performance of the resulting L2NF algorithm (denoted as $\text{L2NF}^{\mathbf{w}=1}_{M=I}$) is 79.4%. This value can be boosted to 84.3% if we adopt the optimized metric $M^\star$ (denoted as $\text{L2NF}^{\mathbf{w}=1}_{M=M^\star}$), and further boosted to 86.6% if we also use the optimal weight vector $\mathbf{w}^\star$ for multimodal fusion (denoted as $\text{L2NF}^{\mathbf{w}=\mathbf{w}^\star}_{M=M^\star}$). As a conclusion, the proposed L2NF framework is able to leverage all the three factors for achieving the state-the-art performance in a systematic and synergic scheme.

5. LIMITATIONS

Despite the promising results on the benchmark search-based face annotation tasks, our work still have some limitations. First, we assume each name corresponds to a unique single person. However, this is not always true for real-life scenarios. For example, it is possible that two persons have the same name or one person may have multiple names. Such kind of practical duplicate name issues may be partially solved by extending our algorithms, e.g., via learning the similarity between any two names both in the name space and visual space. Second, we assume the top retrieved web facial images are related to the query name. This is clearly true for celebrities who have many photos on the internet. 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. Finally, although the performance of L2NF is much better, more facial feature are used which means more computational cost and storage space. We may overcome the limitation by adopting hashing techniques in our further work.

6. CONCLUSIONS

This paper investigated an emerging paradigm of search-based face annotation for automated face naming through mining large-scale web facial images freely available on the WWW. We proposed a novel framework of “Learning to Name Faces" (L2NF) by exploring multi-modal learning on weakly labeled facial image data. In particular, our framework has three major contributions: (i) we suggest enhancing the initial weak labels by a graph-based refinement scheme based on the “label smoothness" assumption; (ii) we propose to explore multiple facial feature representations, and further optimize the distance metric on each facial feature space using distance metric learning techniques; and (iii) finally, we propose to learn the optimal multimodal fusion of diverse facial features by formulating the problem as a learning to rank task, which can be efficiently solved by the existing structural SVM algorithm. We conduct a set of extensive empirical studies on two benchmark real-world facial image databases, in which encouraging results show that the proposed L2NF model significantly boosts the face annotation performance.

Acknowledgements

This research was supported in part by MOE tier 1 project grant (RG33/11), Microsoft Research grant, Interactive and Digital Media Programme Office (IDMPO) and National Research Foundation (NRF) hosted at Media Development Authority (MDA) under Grant No.: MDA/IDM/2012/8/8-2 VOL 01. Jianke Zhu was supported by National Natural Science Foundation of China under the Grant (61103105).

7. REFERENCES

[1] T. Ahonen, A. Hadid, and M. Pietikäinen. Face description with local binary patterns: Application to face recognition. *IEEE TPMAI*, 28(12):2037–2041, 2006.

[2] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[3] 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'04*, 2004.

[4] J. Bu, B. Xu, C. Wu, C. Chen, J. Zhu, D. Cai, and X. He. Unsupervised face-name association via commute distance. In *ACM MM'12*, pages 219–228, 2012.

[5] Z. Cao, Q. Yin, X. Tang, and J. Sun. Face recognition with learning-based descriptor. In *IEEE CVPR'10*, 2010.

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

[7] B.-C. Chen, Y.-Y. Chen, Y.-H. Kuo, and W. H. Hsu. Scalable face image retrieval using attribute-enhanced sparse codewords. *IEEE Trans. on Multimedia*, 2012.

[8] 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 Trans. on Multimedia*, 13, 2011.

[9] P. Duygulu, K. Barnard, J. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV'02*, 2002.

[10] J. Fan, Y. Gao, and H. Luo. Multi-level annotation of natural scenes using dominant image components and semantic concepts. In *ACM MM'04*, pages 540–547, 2004.

[11] S. Gao, I. W.-H. Tsang, and L.-T. Chia. Kernel sparse representation for image classification and face recognition. In *ECCV'10*, ECCV'10, pages 1–14, Berlin, Heidelberg, 2010. Springer-Verlag.

[12] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Automatic face naming with caption-based supervision. In *IEEE CVPR'08*, pages 1–8, 2008.

[13] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Face recognition from caption-based supervision. In *IJCV'12*, 96:64–82, Jan 2012.

[14] M. Guillaumin, J. Verbeek, and C. Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *ECCV'10*, pages 634–647, Sep 2010.

[15] A. Hanbury. A survey of methods for image annotation. *J. Vis. Lang. Comput.*, 19:617–627, October 2008.

[16] S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu. Semi-supervised svm batch mode active learning with applications to image retrieval. *ACM TOIS*, 27(3):1–29, July 2009.

[17] A. Holub, P. Moreels, and P. Perona. Unsupervised clustering for google searches of celebrity images. In *IEEE FG'08*, pages 1 –8, 2008.

[18] 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.

[19] T. Joachims, T. Finley, and C.-N. J. Yu. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, Oct. 2009.

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

[21] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *IEEE ICCV'09*, Oct 2009.

[22] D.-D. Le and S. Satoh. Unsupervised face annotation by mining the web. In *ICDM'08*, pages 383–392, 2008.

[23] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE TIP*, 11(4):467 –476, apr 2002.

[24] A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *ECCV'08*, pages 316–329, 2008.

[25] B. Mcfee and G. Lanckriet. Metric learning to rank. In *ICML'10*, 2010.

[26] T. Mensink and J. J. Verbeek. Improving people search using query expansions. In *ECCV'08*, pages 86–99, 2008.

[27] D. Ozkan and P. Duygulu. A graph based approach for naming faces in news photos. In *IEEE CVPR'06*, pages 1477–1482, 2006.

[28] X. Rui, M. Li, Z. Li, W.-Y. Ma, and N. Yu. Bipartite graph reinforcement model for web image annotation. In *ACM MM'07*, pages 585–594, Augsburg, Germany, 2007.

[29] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, 2008.

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

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

[32] 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*, 2:14:1–14:15, 2011.

[33] J. Zhu, S. C. Hoi, and M. R. Lyu. Face annotation by transductive kernel fisher discriminant. *IEEE TMM*, 10(01):86–96, 2008.

[34] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, September 2005.

[35] C. Wang, F. Jing, L. Zhang, and H.-J. Zhang. Image annotation refinement using random walk with restarts. In *ACM MM'06*, pages 647–650, 2006.

[36] C. Wang, S. Yan, L. Zhang, and H.-J. Zhang. Multi-label sparse coding for automatic image annotation. *IEEE CVPR'09*, 0:1643–1650, 2009.

[37] D. Wang, S. C. Hoi, and Y. He. Mining weakly labeled web facial images for search-based face annotation. In *ACM SIGIR'11*, pages 535–544, 2011.

[38] D. Wang, S. C. H. Hoi, Y. He, and J. Zhu. Retrieval-based face annotation by weak label regularized local coordinate coding. In *ACM MM'11*, pages 353–362, 2011.

[39] X.-J. Wang, L. Zhang, F. Jing, and W.-Y. Ma. Annosearch: Image auto-annotation by search. In *IEEE CVPR'06*, 2006.

[40] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE TPMAI*, 31(2):210–227, April 2008.

[41] F. Wu, Y. Han, Q. Tian, and Y. Zhuang. Multi-label boosting for image annotation by structural grouping sparsity. In *ACM MM'10*, pages 15–24, 2010.

[42] S. C. Hoi and M. R. Lyu. A multimodal and multilevel ranking scheme for large-scale video retrieval. *TMM*, 10(4):607–619, 2008.

[43] Z. Wu, Q. Ke, J. Sun, and H.-Y. Shum. Scalable face image retrieval with identity-based quantization and multi-reference re-ranking. In *IEEE CVPR'10*, pages 3469–3476, 2010.

[44] J. Zhu, S. C. Hoi, and L. V. Gool. Unsupervised face alignment by robust nonrigid mapping. In *ICCV'09*, 2009.

[45] J. Xu, T.-Y. Liu, M. Lu, H. Li, and W.-Y. Ma. Directly optimizing evaluation measures in learning to rank. In *ACM SIGIR'08*, pages 107–114, 2008.

[46] S. C. Hoi, W. Liu, M. R. Lyu, and W.-Y. Ma. Learning distance metrics with contextual constraints for image retrieval. In *CVPR'06*, volume 2, pages 2072–2078, 2006.

[47] L. Yang and R. Jin. Distance metric learning: A comprehensive survey. Technical report, MSU, 2006.

[48] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *ACM SIGIR'07*, pages 271–278, 2007.

[49] L. Wu, S. C. Hoi, R. Jin, J. Zhu, and N. Yu. Distance metric learning from uncertain side information with application to automated photo tagging. In *ACM MM'09*, pages 135–144, 2009.