# A Logical Operator Oriented Face Retrieval Approach: How to Identify a Suspect Using Partial Photo Information from Different Persons?

Yiu-ming Cheung and Zhikai Hu

Department of Computer Science, Hong Kong Baptist University
Hong Kong SAR, China
ymc@comp.hkbu.edu.hk, zkhu94@163.com

**Abstract.** Facial sketch recognition is one of the most commonly used method to identify a suspect when only witnesses are available, which, however, usually leads to four gaps, *i.e.* memory gap, communication gap, description-sketch gap, and sketch-image gap. These gaps limit its application in practice to some extent. To circumvent these gaps, this paper therefore focus on the problem: how to identify a suspect using partial photo information from different persons. Accordingly, we propose a new Logical Operation Oriented Face Retrieval (LOOFR) approach provided that partial information extracted from several different persons' photos is available. The LOOFR defines the new AND and OR operators on these partial information. For example, " eyes of person A AND mouth of person B" means retrieving the target person whose eyes and mouth are similar to that of person A and person B respectively, while "eyes of person A OR eyes of person B" means retrieving target person whose eyes are similar to both person A and B. Evidently, these logical operators cannot be directly implemented by INTERSECTION and UNION in set operations. Meanwhile, they are better for human understanding than set operators. Subsequently, we propose a two-stage LOOFR approach, in which the representations of partial information are learned in the first stage while the logical operations are processed in the second stage. As a result, the target photo of a suspect can be retrieved. Experiments show its promising results.

## 1 Introduction

In suspect identification, one of the most widely used tool is facial sketch recognition technique, as illustrated in Figure 1(a). Normally, a facial sketch recognition method is to sketch the suspect face manually or by computer based on the narrative of a witness, and then tries to match this sketch with the ones in the database such that the suspect is found. In general, such a method needs to address the following hour gaps:

– **Memory gap**. In most cases, a witness only takes a glance at the suspect without seeing him/her clearly, thus resulting in missing some facial details.

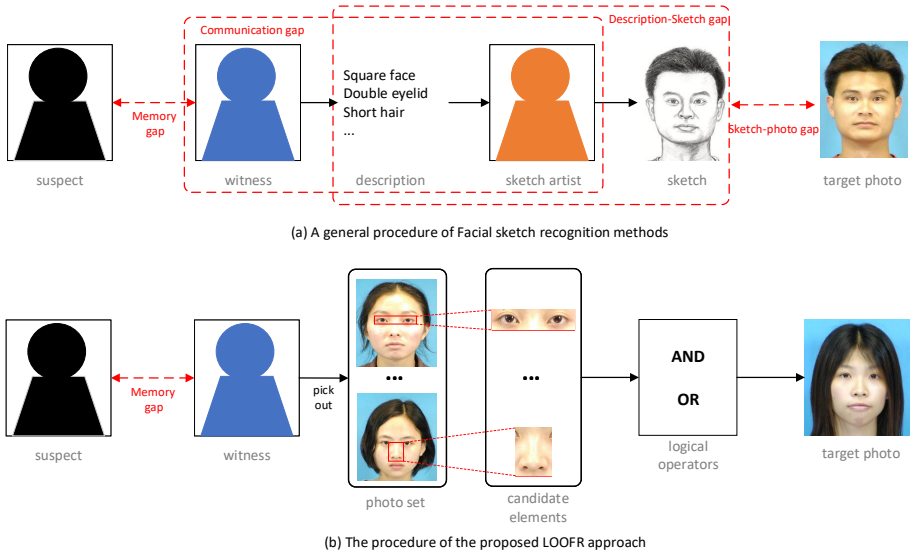


**Fig. 1.** (a) The general procedure of standard facial sketch recognition methods is that, based on a witness' narrative, sketch artist sketches the face of a suspect manually or by computer. Then, the sketch will be compared with the ones in photo database to identify the suspect. (b) The procedure of the proposed LOOFR approach: given a set of photos, a witness can pick out the photos which they think are similar to the suspect and point out which part is the most same; these partial information witnesses select are calculated by logical operators and then matched against photo mugshot database to identify the suspect.

In addition, sometimes they are asked to describe the suspect several days after the event [8, 9], which inevitably increases the uncertainty.

- **Communication gap**. Two persons, *i.e.* the witness and sketch artist, take part in the procedure of facial sketch. In general, even for the same thing, different people may have quite different understanding. It turns out that such imperfect communication between them would affect the quality of sketch [9].
- **Description-sketch gap**. Some descriptions, such as "silent lips" and "murderous eyes", are extremely hard for artists to sketch. This brings another deviation, which is essentially a text-image gap and has been widely studied [20, 13, 14, 22].
- **Sketch-photo gap**. Due to the heterogeneous features of sketches and photos, the spaces they distribute are different. Thus, they cannot be compared directly.

In the literature, a number of methods focusing on bridging the sketch-photo gap have been presented, including feature engineering [2, 18, 3, 5], common space learning [25], and multi-modal learning [31, 30]. Furthermore, Some works,

*e.g.* see [10, 12], have achieved the promising result on the main benchmark [31]. Nevertheless, they have yet to take the memory gap into account. In fact, the difference between the true suspect photo and the sketch in all cases they have tried thus far is relatively small. As far as we know, Uhl et al. [29] firstly attempted to discuss the forensic sketch and tried to bridge the memory gap. Along this line, several papers, *e.g.* see [18, 3], have addressed this problem. Furthermore, some studies [20, 13, 14, 22] have been conducted to build the gap between the text and image, but such work is seldomly applied to the facial sketch recognition task. In addition, to the best of our knowledge, communication and description-sketch gaps have yet to be well explored over the past years.

To alleviate the above-mentioned gaps, as illustrated in Figure 1(b), this paper will consider a scenario below: First, a witness is provided with a set of photos. Then, he/she can select several photos, in which some parts are similar to the suspect, in his/her memory, and points out these parts. Subsequently, a problem is naturally arisen: How to identify the suspect based on these partial information extracted across multiple images? To answer this question, we therefore propose a novel two-stage **Logical Operation Oriented Face Retrieval (LOOFR)** approach, which combines such partial information and matches the combination result with photo mugshot database to identify the suspect. In the LOOFR approach, there are two basic logical operators: **AND** and **OR**, as described below:

- **AND**: When a witness says that the nose and mouth of suspect are similar to two different persons: A and B , respectively, we can use AND to operate them, *i.e.* A's nose AND B's mouth.
- **OR**: When a witness says that the nose of suspect is similar to both A and B, we can use OR to operate them, *i.e.* A's nose OR B's nose.

More complex description, *e.g.* A's nose OR B's nose AND C's eye, can be expressed in terms of these two basic operators. Compared with facial sketch recognition methods, the merits of the proposed approach are at least two-fold:

- The latter three gaps mentioned previously have been bypassed. In the procedure of facial sketch recognition, a witness has to communicate with a sketch artist and misunderstanding often occurs during the communication. On the contrary, in the proposed approach, a witness can complete the whole procedure independently. What they need to do is only to recognize whether the photo is similar to the suspect or not and pick out which part of the photo is similar to the suspect, which therefore circumvent the communication gap and text-image gap. Besides, we directly use photos, not sketch, to retrieve photos, thus the sketch-photo gap vanishes.
- Images are helpful to overcome the memory gap. Instead of recalling the memory on their initiative, a witness is more easily to recall the facial details of a suspect when watching similar photos.

Thus far, there are several related works, *e.g.* multi-query retrieval [17, 1, 7, 34, 32] and instance search [21, 33, 4, 26], but none of them is applicable to

LOOFR problem. In fact, as far as we know, the problem addressed by LOOFR approach has yet to be explored in the literature.

## 2 Related Work

In this section, we review three tasks that are partially similar to LOBFR: facial sketch recognition, multi-query retrieval and instance search.

Facial sketch recognition can be roughly divided into viewed and forensic sketch based face recognition. Viewed sketch means that artists draw the sketch, viewing the corresponding photo. One of the earliest work [28] adopted principal component analysis (PCA) to learning the features and reach a 71% performance on CUHK dataset. Roy et al. [24] employed a fuzzy-based texture encoding model for learning sketch features, but it needs the face of sketch separating from the background. Recently, some works [12, 16, 15] focused on deep learning based recognition framework. Hu et al. [12] fed the sketches of different scales into their multiple input deep networks to learn the effective representation and achieved near-perfect recognition accuracy (99% rank-1) on CUHK dataset. On the contrary, forensic sketch means artists draw the sketch by memory, without the corresponding photo. Uhl et al. [29] is the first one who underlined the importance and challenge of forensic sketch based face recognition. Klare et al. [18] utilized the combination of SIFT and LBP feature to learn a more effective weighting. Later work [3] changed the SIFT and LBEP feature into a new combination of Weber and Wavelet descriptors and reach a better result. Ouyang et al. [23] built a new forensic sketch dataset MGDB to imitate the forgetting process of people. This dataset consists of four kinds of sketchesL viewed, 1-hour, 24-hour and unviewed. Based on this new dataset, they employed a cascade model to overcome the memory gap.

Multi-query retrieval is using multiple sample as query to retrieve the target image. Multiple queries are usually regarded as a method of data augment [1, 6, 7, 17, 32, 34]. Noticing that query samples offered by different users are usually photographed on different view or angle, wang et al. [32] combined a photo of low quality shape with the photos provided by other users who hold the same topic with each others to form a multi-query expansion. These researches are similar to our proposed logical operation OR, where different queries of the same kinds are combined. However, multi-query retrieval based on different queries of different kinds, which is more likely to our proposed logical operation AN, is rarely studied [11, 27]. Hsiao et at. [11] combined the Pareto front method with manifold ranking and proposed a novel method to handle multiple queries of different semantic. Taghizadeh et al. [27] utilised a binary component vector that represents different components of an image to handle multiple queries retrieval.

Instance search (INS) [4, 21, 26, 33] is using a query image of a specific instance to retrieve images containing that instance. It is quite similar to our task that using local information, such as eyes or mouth, to retrieve the target face whose eyes or mouth are similar to the query sample. Yu et al. [33] proposed a Fuzzy Objects Matching (FOM) framework to explore the similarity between

the query sample and images in the dataset and used object proposals to detect whether images contain the potential regions of the query sample. Song et al. [26] combined the deep networks with hash method to cope with large scale instance search problem. They learned two kinds of hash codes: global region hash codes and local region hash codes. The query sample should be first compared with global region hash codes to get a rank, then be compared with local region hash codes to re-rank the former rank.

## 3 Proposed Approach

Suppose that a face dataset $\mathbf{F} = [\mathbf{f}_1, ..., \mathbf{f}_n]$ contains $n$ face samples and each face $\mathbf{f}_j$ consists of five elements $\{\mathbf{e}_j^{(1)}, \mathbf{e}_j^{(2)}, \mathbf{e}_j^{(3)}, \mathbf{e}_j^{(4)}, \mathbf{e}_j^{(5)}\}$, which represent eyes, eyebrow, nose, mouth and outline of face respectively. Same elements of all faces constitute new datasets $\mathbf{E}^{(i)} = [\mathbf{e}_1^{(i)} ..., \mathbf{e}_n^{(i)}] \in \mathbb{R}^{d_i \times n}, i = 1, 2, 3, 4, 5$, where $d_i$ is the dimension of the $i$-th element. Our aim is to retrieve target face via two basic logical operators **AND** and **OR**. That is:

1. Use $\mathbf{e}_i^{(1)}$ and $\mathbf{e}_j^{(2)}$ (**AND** operation) for retrieving $\mathbf{f}_k$ whose $\mathbf{e}_k^{(1)}$ is similar to $\mathbf{e}_i^{(1)}$ and $\mathbf{e}_k^{(2)}$ is similar to $\mathbf{e}_j^{(2)}$.
2. Use $\mathbf{e}_i^{(1)}$ and $\mathbf{e}_j^{(1)}$ (**OR** operation) for retrieving $\mathbf{f}_k$ whose $\mathbf{e}_k^{(1)}$ is similar to both $\mathbf{e}_i^{(1)}$ and $\mathbf{e}_j^{(1)}$.

To tackle this problem, we propose a two-stage approach: in the first stage, we learn five sparse hashing codes, or representations of elements, $\mathbf{Z}^{(i)} \in \{0, 1\}^{k_i \times n}$ and their corresponding dictionaries $\mathbf{D}^{(i)} = [\mathbf{d}_1^{(i)} ..., \mathbf{d}_{k_i}^{(i)}] \in \mathbb{R}^{d_i \times k_i}$ and projections $\mathbf{P}^{(i)} \in \mathbb{R}^{k_i \times d_i}$ for all elements respectively, where $k_i$ is the code length of $Z^{(i)}$; in the second stage, giving query samples and corresponding elements, we will get several candidate sets via learned $\mathbf{Z}^{(i)}$ and $\mathbf{P}^{(i)}$. Then, logical operation is performed on these candidate sets to make the final decision.

### 3.1 Dictionary Learning for Representations of Five Facial Elements

For $i$-th element $\mathbf{E}^{(i)}$, we aim to learn an enriched dictionary $\mathbf{D}^{(i)}$ and a sparse coefficient matrix $\mathbf{Z}^{(i)}$. Specifically, we minimize the following object function:

$$\begin{aligned} \mathcal{L}_1^{(i)} &= ||\mathbf{E}^{(i)} - \mathbf{D}^{(i)}\mathbf{Z}^{(i)}||_F^2 + \gamma||\mathbf{Z}^{(i)}||_1 \\ s.t. \quad & \forall j,\ {\mathbf{d}_j^{(i)}}^T \mathbf{d}_j^{(i)} = 1, \end{aligned} \tag{1}$$

where $\lambda > 0$ is a trade-off parameter.

For similar elements $\mathbf{e}_j^{(i)}$ and $\mathbf{e}_k^{(i)}$, their corresponding sparse codes $\mathbf{z}_j^{(i)}$ and $\mathbf{z}_k^{(i)}$ should be as same as possible. To this end, we first define an affinity matrix $\mathbf{S}^{(i)} \in \mathbb{R}^{n \times n}$ by

$$\mathbf{s}_{jk}^{(i)} = \begin{cases} 1, if\ \mathbf{e}_j^{(i)}\ and\ \mathbf{e}_k^{(i)}\ are\ similar, \\ 0, \qquad\qquad otherwise, \end{cases} \tag{2}$$

where $\mathbf{s}_{jk}^{(i)}$ is the $j$-th row and $k$-th column of $\mathbf{S}^{(i)}$. Then, we minimize the following object function:

$$\begin{aligned}\mathcal{L}_2^{(i)} &= \sum_{j=1}^{n}\sum_{k=1}^{n}\mathbf{s}_{jk}^{(i)}||\mathbf{z}_j^{(i)} - \mathbf{z}_k^{(i)}||_2^2 \\ &= Tr(\mathbf{Z}^{(i)}(\mathbf{G}^{(i)} - \mathbf{S}^{(i)})\mathbf{Z}^{(i)^T}) \\ &= Tr(\mathbf{Z}^{(i)}\mathbf{L}^{(i)}\mathbf{Z}^{(i)^T}),\end{aligned} \tag{3}$$

where $\mathbf{G}^{(i)} \in \mathbb{R}^{n\times n}$ is a diagonal matrix whose entries are the column sum of $\mathbf{S}^{(i)}$, *i.e.* $\mathbf{g}_{kk}^{(i)} = \sum_j \mathbf{s}_{jk}^{(i)}$ and $\mathbf{L}^{(i)} = \mathbf{G}^{(i)} - \mathbf{S}^{(i)}$.

For out of samples, we prefer to learn a projection $\mathbf{P}^{(i)}$ to directly map them into sparse codes rather than solve sparse codes via Eq.(1). Subsequently, we minimize the following object function:

$$\mathcal{L}_3^{(i)} = \beta||\mathbf{Z}^{(i)} - \mathbf{P}^{(i)}\mathbf{E}^{(i)}||_F^2 + \lambda||\mathbf{P}^{(i)}||_F^2, \tag{4}$$

where $\beta$ and $\lambda$ are trade-off parameters and $||\mathbf{P}^{(i)}||_F^2$ is a regularization term.

To be concise, we omit the superscript of the letters in the subsequent equations. For each element, we get the following overall loss function

$$\begin{aligned}\mathcal{L} =& ||\mathbf{E} - \mathbf{D}\mathbf{Z}||_F^2 + \gamma||\mathbf{Z}||_1 + \alpha Tr(\mathbf{Z}\mathbf{L}\mathbf{Z}^T) \\ &+ \beta||\mathbf{Z} - \mathbf{P}\mathbf{E}||_F^2 + \lambda||\mathbf{P}||_F^2 \\ &s.t.\quad \forall j,\ \mathbf{d}_j^T\mathbf{d}_j = 1.\end{aligned} \tag{5}$$

Because of the term $||\mathbf{Z}||_1$, $\mathcal{L}$ is not differentiable for $\mathbf{Z}$ in the whole field of real numbers. In general, Least Absolute Shrinkage and Selection Operator (Lasso) or K-SVD are adopted to update $\mathbf{Z}$ bit by bit or column by column. In this paper, we combine dictionary learning with hashing method and introduce a new constraint $\mathbf{Z} \in [0,1]^{k\times n}$, aiming to use binary codes to represent $\mathbf{Z}$. Then, Eq.(5) becomes

$$\begin{aligned}\mathcal{L} =& ||\mathbf{E} - \mathbf{D}\mathbf{Z}||_F^2 + \gamma||\mathbf{Z}||_1 + \alpha Tr(\mathbf{Z}\mathbf{L}\mathbf{Z}^T) \\ &+ \beta||\mathbf{Z} - \mathbf{P}\mathbf{E}||_F^2 + \lambda||\mathbf{P}||_F^2 \\ &s.t.\quad \forall j,\ \mathbf{d}_j^T\mathbf{d}_j = 1, \mathbf{Z} \in [0,1]^{k\times n}.\end{aligned} \tag{6}$$

Compared with Eq.(5), there are two merits to optimize the loss function in Eq.(6):

1. It is easier to solve Eq.(6) than Eq.(5). The loss function is continue and differentiable for $\mathbf{Z}$ in the interval $\mathbf{Z} \in [0,1]^{k\times n}$, thus, we can use gradient descent or least squares method to solve it; while several cusps exist when $\mathbf{Z} \in \mathbb{R}^{k\times n}$, these methods are not suitable.
2. Optimizing Eq.(6) is time-saving. Whole $\mathbf{Z}$ can be updated simultaneously by gradient descent or least square method, while it should be updated bit by bit or column by column via Lasso or K-SVD, which means more iteration and computation.

The minimization problem in Eq.(6) can be solved by alternating optimization:

**D-step**: Fix $\mathbf{Z}$ and $\mathbf{P}$, we can update the dictionary $\mathbf{D}$ column by column. Let $\mathbf{D} = [\mathbf{d}_1, ..., \mathbf{d}_k]$ and $\mathbf{Z} = [\mathbf{z}_1; ...; \mathbf{z}_k]$, where $\mathbf{d}_i$ and $\mathbf{z}_j$ are the $i$-th column of $\mathbf{D}$ and $j$-th row of $\mathbf{Z}$ respectively.

When update $\mathbf{d}_i$, the other columns of $\mathbf{D}$ are constant and Eq.(5) can be rewritten to

$$\mathcal{L} = ||\hat{\mathbf{E}} - \mathbf{d}_i\mathbf{z}_i||_F^2 + const, \tag{7}$$

where $\hat{\mathbf{E}} = \mathbf{E} - \sum_{j \neq i} \mathbf{d}_j\mathbf{z}_j$.

**Z-step**: Fix $\mathbf{D}$ and $\mathbf{P}$, loss function in Eq.(6) is differentiable for $\mathbf{Z}$. In addition, $\mathcal{L}$ is bounded in the closed interval $\mathbf{Z} \in [0, 1]^{k \times n}$. Thus, $\mathcal{L}$ must attain its minimum at the point where $\frac{\partial \mathcal{L}}{\partial \mathbf{Z}} = 0$ or the boundary of the closed interval. Let $\frac{\partial \mathcal{L}}{\partial \mathbf{Z}} = 0$, we can solve $\mathbf{Z}$ by

$$\mathbf{AZ} + \mathbf{ZB} + \mathbf{C} = 0, \tag{8}$$

where $\mathbf{A} = 2\mathbf{D}^T\mathbf{D}$, $\mathbf{B} = 2(\alpha\mathbf{L} + \beta\mathbf{I})$ and $\mathbf{C} = \gamma\mathbf{1} - 2(\mathbf{D}^T + \beta\mathbf{P})\mathbf{E}$ and $\mathbf{1}$ denotes the matrix whose elements are all ones and $\mathbf{I}$ is an identity matrix. Eq.(8) is a Sylvester equation [19], which can be solved by the lyap function of MATLAB.

If the solution of Eq.(8) is out of the interval $[0, 1]^{k \times n}$, we use

$$\mathbf{z}_{ij}^* = \begin{cases} 0 & , if\ \mathbf{z}_{ij} < 0, \\ \mathbf{z}_{ij} & , if\ \mathbf{z}_{ij} \in [0, 1], \\ 1 & , if\ \mathbf{z}_{ij} > 1. \end{cases} \tag{9}$$

to modify its coordinates out of $[0, 1]$ to the boundary and keep grads of other directions being 0. In this way, we ensure that the solution of Eq.(8) locates at the point where $\frac{\partial \mathcal{L}}{\partial \mathbf{Z}} = 0$ or the boundary of $[0, 1]^{k \times n}$.

**P-step**: Fix $\mathbf{D}$ and $\mathbf{Z}$ and let $\frac{\partial \mathcal{L}}{\partial \mathbf{P}} = 0$, we can get

$$\mathbf{P} = \mathbf{ZE}^T(\mathbf{EE}^T + \frac{\lambda}{\beta}\mathbf{I})^{-1}, \tag{10}$$

where $\mathbf{I}$ is an identity matrix.

### 3.2 Logical Operation Oriented Face Retrieval

**AND**—Given query samples $\mathbf{q}_1^{(i)}$ and $\mathbf{q}_2^{(j)}$ of elements $\mathbf{E}^{(i)}$ and $\mathbf{E}^{(j)}$ respectively, we can compute their corresponding sparse codes by

$$\hat{\mathbf{z}}_1^{(i)} = \mathbf{P}^{(i)}\mathbf{q}_1^{(i)}\ ,\ \hat{\mathbf{z}}_2^{(j)} = \mathbf{P}^{(j)}\mathbf{q}_2^{(j)}. \tag{11}$$

*Set operation*: Comparing $\hat{\mathbf{z}}_1^{(i)}$ and $\hat{\mathbf{z}}_2^{(j)}$ with $\mathbf{Z}^{(i)}$ and $\mathbf{Z}^{(j)}$, respectively, we can get two top-K nearest neighbor candidate sets $\mathcal{A}$ and $\mathcal{B}$, which contain the indices of the top-K nearest samples in $\mathbf{Z}^{(i)}$ and $\mathbf{Z}^{(j)}$. Then, the final decision is made by $\mathcal{C}_1 = \mathcal{A} \cap \mathcal{B}$.

*Our approach*: We concatenate $\hat{\mathbf{z}}_1^{(i)}$ and $\hat{\mathbf{z}}_2^{(j)}$, $\mathbf{Z}^{(i)}$ and $\mathbf{Z}^{(j)}$ to get a new query and retrieval set

$$\hat{\mathbf{z}} = [\hat{\mathbf{z}}_1^{(i)}; \hat{\mathbf{z}}_2^{(j)}] \ , \ \mathbf{Z} = [\mathbf{Z}^{(i)}; \mathbf{Z}^{(j)}]. \tag{12}$$

Then, we compare $\hat{\mathbf{z}}$ with $\mathbf{Z}$ to get top-K nearest neighbor result $\mathcal{C}_2$.

**OR**—Given query samples $\mathbf{q}_1^{(i)}$ and $\mathbf{q}_2^{(i)}$ of elements $\mathbf{E}^{(i)}$, we can compute their corresponding sparse codes by

$$\hat{\mathbf{z}}_{1,2}^{(i)} = \mathbf{P}^{(i)}\mathbf{q}_{1,2}^{(i)}. \tag{13}$$

*Set operation*: Comparing $\hat{\mathbf{z}}_{1,2}^{(i)}$ with $\mathbf{Z}^{(i)}$, respectively, we can get two top-K nearest neighbor candidate sets $\mathcal{A}_1$ and $\mathcal{A}_2$, which contain the indices of the top-K nearest samples in $\mathbf{Z}^{(i)}$. Then, the final decision is made by $\mathcal{C}_1 = \mathcal{A}_1 \cup \mathcal{A}_2$.

*Our approach*: We compare $\hat{\mathbf{z}}_{1,2}^{(i)}$ with $\mathbf{Z}^{(i)}$ simultaneously and record the distance between $\hat{\mathbf{z}}_{1,2}^{(i)}$ and each item of $\mathbf{Z}^{(i)}$. Based on these records, we get the top-K nearest neighbor result $\mathcal{C}_2$.

**AND+OR**—Given query samples $\mathbf{q}_1^{(i)}$ and $\mathbf{q}_2^{(i)}$ of elements $\mathbf{E}^{(i)}$, and $\mathbf{q}_3^{(j)}$ of elements $\mathbf{E}^{(j)}$, we can compute their corresponding sparse codes by

$$\hat{\mathbf{z}}_{1,2}^{(i)} = \mathbf{P}^{(i)}\mathbf{q}_{1,2}^{(i)} \ , \ \hat{\mathbf{z}}_3^{(j)} = \mathbf{P}^{(j)}\mathbf{q}_3^{(j)}. \tag{14}$$

*Set operation*: Comparing $\hat{\mathbf{z}}_{1,2}^{(i)}$ and $\hat{\mathbf{z}}_3^{(2)}$ with $\mathbf{Z}^{(i)}$ and $\mathbf{Z}^{(j)}$, respectively, we can get three top-K nearest neighbor candidate sets $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{B}$, which contain the indices of the top-K nearest samples in $\mathbf{Z}^{(i)}$ and $\mathbf{Z}^{(j)}$. Then, the final decision is made by $\mathcal{C}_1 = (\mathcal{A}_1 \cup \mathcal{A}_2) \cap \mathcal{B}$.

*Our approach*: We concatenate $\hat{\mathbf{z}}_1^{(i)}$ and $\hat{\mathbf{z}}_2^{(j)}$, $\mathbf{Z}^{(i)}$ and $\mathbf{Z}^{(j)}$ to get a new query and retrieval set

$$\begin{gathered} \hat{\mathbf{z}}_{\mathbf{1}} = [\hat{\mathbf{z}}_1^{(i)}; \hat{\mathbf{z}}_3^{(j)}] \ , \ \hat{\mathbf{z}}_{\mathbf{2}} = [\hat{\mathbf{z}}_2^{(1)}; \hat{\mathbf{z}}_3^{(j)}] \\ \mathbf{Z} = [\mathbf{Z}^{(i)}; \mathbf{Z}^{(j)}]. \end{gathered} \tag{15}$$

Then, we compare $\hat{\mathbf{z}}_{1,2}$ with $\mathbf{Z}$ simultaneously and record the distance between $\hat{\mathbf{z}}_{1,2}$ and each item of $\mathbf{Z}$. Based on these records, we get the top-K nearest neighbor result $\mathcal{C}_2$.

## 4 Experiments

### 4.1 Dataset and Performance Measurement

CUHK student data set [31] consists of 188 faces. We first cut them into five elements: eye, eyebrow, nose, mouth and outline, and then annotate the similarity of the samples of the same element. Each element is represented by a 128-d SIFT feature. 170 faces and their corresponding elements are randomly selected to form the training set, and remaining is the test set.

Two criteria, R@k and Average Index, are adopted to measure the performance of our proposed method. R@k is the accuracy of top-k rank retrieval result. Average Index denotes the average place where the target face appears in the results.

### 4.2 Results of AND Operation

**Table 1.** The results of R@k (k=10,20,50) of the three operations: use only one element ($\mathcal{A}$ and $\mathcal{B}$), set operation ($\mathcal{C}_1$) and our strategy AND ($\mathcal{C}_2$). Best results are marked in bold.

| element | | R@10 | | | | R@20 | | | | R@50 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{q}_1^{(1)}$ | $\mathbf{q}_2^{(2)}$ | $\mathcal{A}$ | $\mathcal{B}$ | $\mathcal{C}_1$ | $\mathcal{C}_2$ | $\mathcal{A}$ | $\mathcal{B}$ | $\mathcal{C}_1$ | $\mathcal{C}_2$ | $\mathcal{A}$ | $\mathcal{B}$ | $\mathcal{C}_1$ | $\mathcal{C}_2$ |
| eye | eyebrow | 9.7 | **10.2** | 0.2 | 10.0 | 15.9 | 17.0 | 2.2 | **19.2** | 36.1 | 36.5 | 12.4 | **41.4** |
| eye | mouth | 8.7 | 7.4 | 0.4 | **9.3** | 16.5 | 15.3 | 1.8 | **17.3** | 37.8 | 37.6 | 13.9 | **40.2** |
| eye | nose | **11.6** | 7.9 | 1.1 | **11.6** | 16.1 | 18.0 | 3.3 | **18.2** | 36.3 | 35.4 | 13.8 | **38.3** |
| eye | outline | 8.9 | 8.1 | 1.1 | **9.2** | **17.6** | 13.4 | 2.7 | 16.6 | 35.9 | 34.9 | 13.4 | **37.2** |
| eyebrow | mouth | 9.4 | 9.6 | 0.4 | **11.5** | 16.8 | 18.2 | 3.8 | **18.7** | 34.8 | 37.9 | 13.8 | **39.0** |
| eyebrow | nose | 11.8 | 12.5 | 2.1 | **12.9** | 18.5 | 21.1 | 4.0 | **22.4** | 38.7 | 37.7 | 14.1 | **44.2** |
| eyebrow | outline | **9.5** | 6.8 | 1.6 | **9.5** | 15.5 | 13.4 | 2.7 | **16.8** | 30.9 | **34.9** | 13.1 | 32.6 |
| mouth | nose | 7.6 | 11.1 | 0.9 | **11.2** | 16.2 | **19.3** | 4.0 | **19.3** | 37.1 | 37.1 | 15.4 | **40.8** |
| mouth | outline | 7.6 | 5.7 | 0.4 | **8.7** | **17.6** | 12.6 | 2.5 | 15.3 | 37.9 | 37.4 | 13.7 | **39.4** |
| nose | outline | **9.3** | 6.7 | 0.8 | 8.9 | **17.0** | 14.2 | 3.3 | 16.2 | 36.2 | 34.6 | 12.0 | **37.7** |

The results of logical operation AND are reported in Table 1. It can be observed that: firstly, in most cases (*i.e.* 24/30), our proposed approach ($\mathcal{C}_2$) achieves a better result than both using only one element and set operation; secondly, results of intersect operation ($\mathcal{C}_1$) are always worse than that of using only one element. This is determined by the property of set operation: the intersection of two sets is always smaller than them. The worse performance of intersect operation means that two elements usually do not get the right answer simultaneously.

In contrast, our approach takes the complementary advantages of both elements. For example, when a high-ranked result is returned by querying one element and a low-ranked result is returned by querying another element, our proposed AND operation makes the result returned by using both of them reach a moderate place.

### 4.3 Results of OR Operation

The results of logical operation OR are reported in Table 2. It can be observed that: firstly, in most cases (*i.e.* 10/15), our proposed logical operation OR ($\mathcal{C}_2$) achieves a better result than using only one element; secondly, contrary to intersect operation, results of union operation ($\mathcal{C}_1$) are always better than that of other three operations. This is also determined by the property of set operation: the union of two sets is always bigger than them.

Although this result is impressive, it does not means that the set operation is superior to our approach. Because the account of $\mathcal{C}_1$ do not represent the data that similar to both queries. Take eye OR eye for example, the percentage of this part of data only accounts for (38.5+40.5-60.6=)18.4%, less than the 1/3 of

**Table 2.** The results of R@k (k=10,20,50) of the three operations: use only one element ($\mathcal{A}_1$ and $\mathcal{A}_2$), set operation ($\mathcal{C}_1$) and our strategy OR ($\mathcal{C}_2$). Best results are marked in bold.

| element | | R@10 | | | | R@20 | | | | R@50 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{q}_1^{(1)}$ | $\mathbf{q}_2^{(1)}$ | $\mathcal{A}_1$ | $\mathcal{A}_2$ | $\mathcal{C}_1$ | $\mathcal{C}_2$ | $\mathcal{A}_1$ | $\mathcal{A}_2$ | $\mathcal{C}_1$ | $\mathcal{C}_2$ | $\mathcal{A}_1$ | $\mathcal{A}_2$ | $\mathcal{C}_1$ | $\mathcal{C}_2$ |
| eye | eye | 8.7 | 8.4 | **16.4** | 7.4 | 15.7 | 18.1 | **30.8** | 18.7 | 38.5 | 40.5 | **60.5** | 40.1 |
| eyebrow | eyebrow | 9.7 | 8.2 | **16.5** | 12.0 | 19.1 | 16.9 | **30.3** | 20.2 | 36.7 | 41.2 | **62.2** | 43.8 |
| mouth | mouth | 9.6 | 11.4 | **20.3** | 10.7 | 17.0 | 18.5 | **32.5** | 19.9 | 38.7 | 36.9 | **60.9** | 42.1 |
| nose | nose | 13.4 | 7.9 | **19.1** | 11.8 | 19.6 | 14.7 | **30.6** | 21.5 | 43.5 | 36.6 | **63.4** | 44.8 |
| outline | outline | 7.1 | 7.8 | **14.3** | 6.9 | 13.3 | 15.0 | **24.5** | 16.2 | 30.9 | 34.0 | **52.5** | 34.9 |

the result reported in Table 1. Thus, the impressive results of union operation in Table 1 are of little significance and such plenitude vanishes when conducting a more complex logical operation which we discuss in the next section.

### 4.4 Results of AND+OR Operation

**Table 3.** The results of R@k (k=20,50) of the three operations: use only one element ($\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{B}$), set operation ($\mathcal{C}_1$) and our approach AND+OR ($\mathcal{C}_2$). Best results are marked in bold. Better results that logical operation OR achieves than only one element does are underlined.

| element | | | R@20 | | | | | R@50 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{q}_1^{(i)}$ | $\mathbf{q}_2^{(i)}$ | $\mathbf{q}_3^{(2)}$ | $\mathcal{A}_1$ | $\mathcal{A}_2$ | $\mathcal{B}$ | $\mathcal{C}_1$ | $\mathcal{C}_2$ | $\mathcal{A}_1$ | $\mathcal{A}_2$ | $\mathcal{B}$ | $\mathcal{C}_1$ | $\mathcal{C}_2$ |
| eye | eye | eyebrow | 11.5 | 10.5 | **16.0** | 1.6 | 14.6 | 32.8 | 29.0 | 34.1 | 17.0 | **39.6** |
| eye | eye | mouth | 9.6 | 13.2 | 18.6 | 3.5 | **19.9** | 29.8 | 32.1 | **39.1** | 18.2 | 38.8 |
| eye | eye | nose | 13.8 | 13.2 | 19.2 | 5.0 | **20.9** | 32.0 | 31.3 | 35.5 | 19.1 | **36.4** |
| eye | eye | outline | 10.9 | 13.8 | 11.6 | 2.2 | **15.2** | 29.8 | 32.9 | **37.1** | 19.6 | 34.5 |
| eyebrow | eyebrow | eye | 15.5 | 16.6 | 12.1 | 3.0 | **17.1** | 33.4 | 35.4 | 31.2 | 17.8 | **39.6** |
| eyebrow | eyebrow | mouth | 15.4 | 15.2 | **21.0** | 4.8 | 19.6 | 30.6 | 32.8 | 36.1 | 16.9 | **40.3** |
| eyebrow | eyebrow | nose | 15.5 | 14.1 | 14.8 | 3.9 | **16.0** | 34.4 | 31.8 | **39.0** | 20.3 | 37.1 |
| eyebrow | eyebrow | outline | 18.5 | 16.8 | 18.5 | 6.3 | **22.4** | 38.5 | 34.6 | 40.1 | 25.3 | **42.0** |
| mouth | mouth | eye | 11.9 | 14.6 | 13.7 | 3.4 | **17.7** | 31.6 | 33.1 | 32.2 | 18.9 | **37.5** |
| mouth | mouth | eyebrow | **17.6** | 14.2 | 15.9 | 5.8 | 16.7 | 36.7 | 33.8 | 32.1 | 19.3 | **39.8** |
| mouth | mouth | nose | 17.1 | 14.2 | 18.6 | 6.8 | **20.7** | 37.6 | 34.0 | 40.8 | 25.5 | **44.9** |
| mouth | mouth | outline | 13.4 | 14.0 | 16.1 | 4.0 | **17.4** | 33.6 | 34.9 | 35.6 | 20.4 | **39.2** |
| nose | nose | eye | 15.5 | 16.5 | 17.1 | 5.9 | **18.1** | 30.5 | 34.3 | 34.3 | 19.8 | **39.0** |
| nose | nose | eyebrow | 12.8 | 12.3 | 17.7 | 4.2 | **18.0** | 32.8 | 37.6 | 35.9 | 21.9 | **37.8** |
| nose | nose | mouth | 14.9 | 15.0 | 18.6 | 5.1 | **20.3** | 34.2 | 35.8 | 42.0 | 23.7 | **45.8** |
| nose | nose | outline | **14.3** | 12.7 | 10.5 | 3.9 | 12.7 | 30.8 | 32.5 | **34.1** | 16.8 | 32.8 |
| outline | outline | eye | 9.2 | 11.5 | **14.0** | 2.2 | 12.4 | 26.8 | 37.1 | **37.6** | 18.8 | 35.7 |
| outline | outline | eyebrow | 13.4 | 14.0 | 13.9 | 4.5 | **15.6** | 33.1 | **35.7** | 32.2 | 18.3 | 34.9 |
| outline | outline | mouth | 10.5 | 13.6 | 15.4 | 3.1 | **16.7** | 28.1 | 33.5 | **38.4** | 17.9 | 37.5 |
| outline | outline | nose | 10.9 | 11.2 | **13.7** | 1.5 | 10.5 | 29.8 | **32.8** | 31.3 | 14.9 | 31.5 |

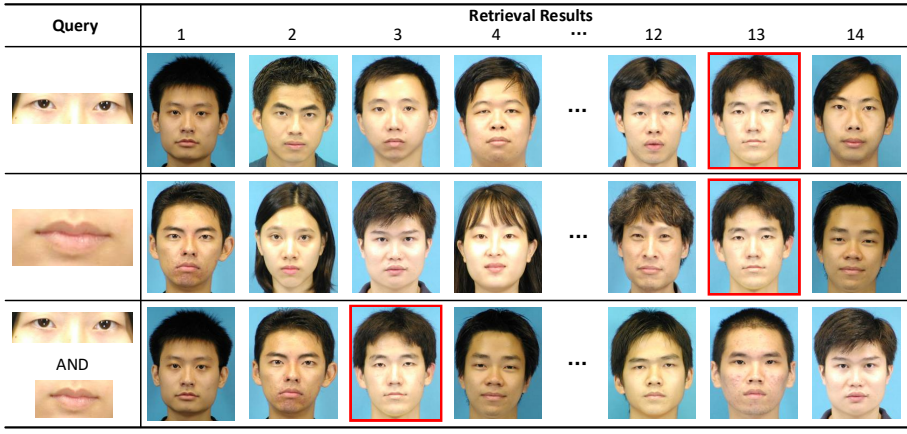


**Fig. 2.** When using only one element (eye or mouth), the target face (in bold border) is ranked at the 13th; when using eye AND mouth, the target face is re-ranked at the 3rd.

To further explore the more complicated application of logical operation, we conducted the experiment of AND+OR, the results are reported in Table 3.

It can be observed that in most cases (*i.e.* 26/40), the proposed approach ($\mathcal{C}_2$) achieves better performance, which are in coordinated with the former results. This demonstrates the effectiveness of the proposed approach of logical operation.

As mentioned before, the impressive results of union operation ($\mathcal{C}_1$) in Table 1 do not appear again. A plausible reason is that the result of set operation is determined by $\mathcal{C}_1 = (\mathcal{A}_1 \cup \mathcal{A}_2) \cap \mathcal{B}$, which equals to $\mathcal{C}_1 = (\mathcal{A}_1 \cap \mathcal{B}) \cup (\mathcal{A}_2 \cap \mathcal{B})$. From Table 1, it is easy to find that intersect operation dramatically decreases the retrieval results. This means that the size of two sets $(\mathcal{A}_1 \cap \mathcal{B})$ and $(\mathcal{A}_2 \cap \mathcal{B})$ in the latter equation are extremely small. Even the union operation, which achieves "considerable results" in Table 1, cannot increase the result by a big margin.

### 4.5 Rank Improvement after Logical Operation

The purpose of logical operation is to use more partial information to get the result with more confidence (*i.e.* at higher rank). For example, as shown in Figure 2, when using only one element (eye or mouth), the target face (in bold border) is ranked at the 13th; when using eye AND mouth, the target face is re-ranked at the 3rd.

Apparently, set operation violates this need, because set is unordered, rank information was discarded when conducting the set operation. Thus logical operation cannot directly implemented by set operation. In contrast, as shown in Table 4, in most cases, logical operations AND (*i.e.* 8/10) and OR (*i.e.* 5/5)

**Table 4.** The Average index of the tow operations: use only one element ($\mathcal{A}$ and $\mathcal{B}$) and our approach AND ($\mathcal{C}_2$). Best results are marked in bold.

| | element | | Average Index | | |
|---|---|---|---|---|---|
| | $\mathbf{q}_1^{(i)}$ | $\mathbf{q}_2^{(j)}/\mathbf{q}_2^{(i)}$ | $\mathcal{A}/\mathcal{A}_1$ | $\mathcal{B}/\mathcal{A}_2$ | $\mathcal{C}_2$ |
| AND | eye | eyebrow | 77 | 77 | **74** |
| | eye | mouth | 76 | 76 | **73** |
| | eye | nose | 76 | 77 | **74** |
| | eye | outline | **76** | 81 | 77 |
| | eyebrow | mouth | 80 | 77 | **76** |
| | eyebrow | nose | 73 | 72 | **68** |
| | eyebrow | outline | 82 | **80** | 81 |
| | mouth | nose | 76 | 77 | **74** |
| | mouth | outline | 76 | 80 | **75** |
| | nose | outline | 77 | 80 | **75** |
| OR | eye | eye | 75 | 73 | **71** |
| | eyebrow | eyebrow | 74 | 75 | **71** |
| | mouth | mouth | 73 | 71 | **67** |
| | nose | nose | 70 | 75 | **68** |
| | outline | outline | 83 | 81 | **80** |

get smaller average indexes, which means it indeed improves the retrieval performance and returns the results in a better rank with a stronger confidence.

To further analyze the effect of logical operation on the rank of results, we count the number of index that denotes the place where the target face appears in the results of different intervals I-IX (representing [1, 20], [21, 40], [41, 60], [61, 80], [81,100], [101,120], [121, 140], [141, 160] and [161, 170]). The smaller the index is, the higher confidence level of the result obtains. Some results are represented in Figure 3.

It is clear that, compared with only using one element, logical operations increase the number of index falling to the intervals I, II and III, and decrease that to the intervals VII, VIII and IX. That is, logical operation re-rank the target face in a higher order by using multiple partial information. This accounts for the results in Table 4.

## 5 Conclusion

This paper has addressed the problem of identifying a suspect using partial photo information from different persons. Accordingly, we have proposed the novel LOOFR approach to bypass the thorny problems faced by the existing facial sketch recognition methods. In this two-stage approach, representations of elements are learned in the first stage, and logical operators: AND and OR, are utilized on representations in the second stage to retrieve target face with a better rank and stronger confidence. We have conducted several experiments on three scenarios AND, OR and their combination AND+OR and compared

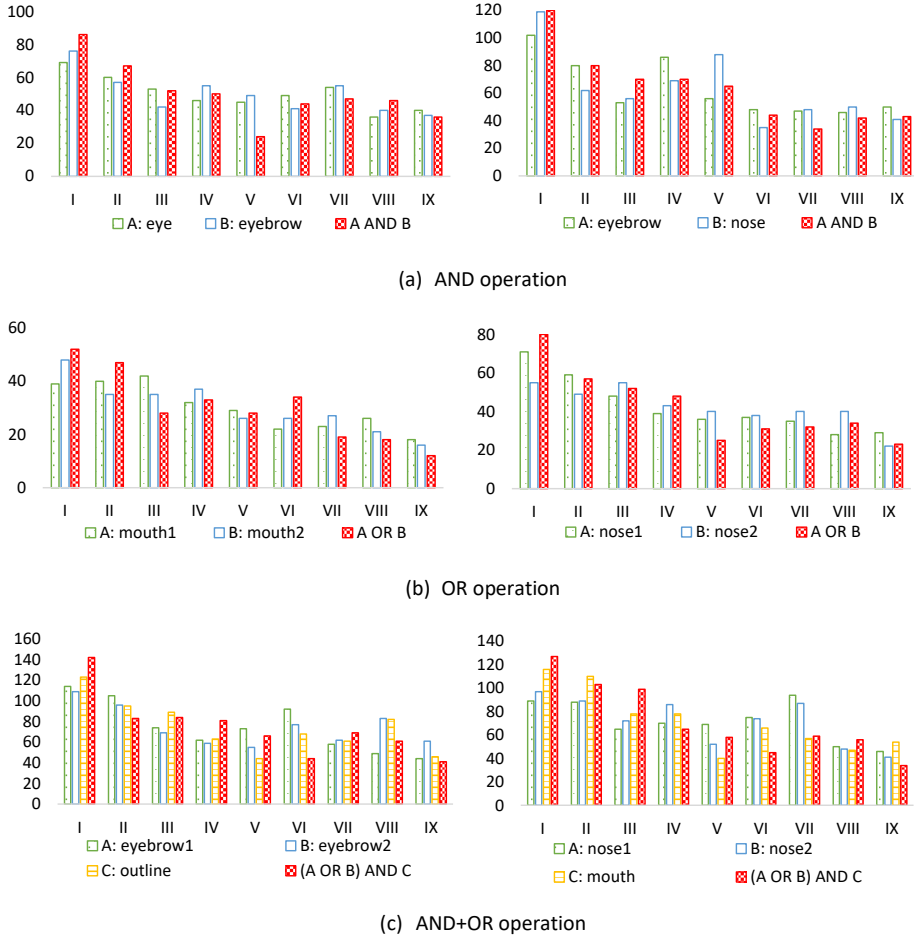


**Fig. 3.** The counts of rank of target face in results falling into different intervals I-IX, which representing [1, 20], [21, 40], [41, 60], [61, 80], [81,100], [101,120], [121, 140], [141, 160] and [161, 170] respectively.

our approach with set operation. The results have shown the effectiveness of the proposed approach.

## References

1. Arandjelović, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2911–2918. IEEE (2012)
2. Bhatt, H.S., Bharadwaj, S., Singh, R., Vatsa, M.: On matching sketches with digital face images. In: 2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS). pp. 1–7. IEEE (2010)

3. Bhatt, H.S., Bharadwaj, S., Singh, R., Vatsa, M.: Memetically optimized mcwld for matching sketches with digital face images. IEEE Transactions on Information Forensics and Security **7**(5), 1522–1535 (2012)
4. Bhattacharjee, S.D., Yuan, J., Huang, Y., Meng, J., Duan, L.: Query adaptive multiview object instance search and localization using sketches. IEEE Transactions on Multimedia **20**(10), 2761–2773 (2018)
5. Choi, J., Sharma, A., Jacobs, D.W., Davis, L.S.: Data insufficiency in sketch versus photo face recognition. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. pp. 1–8. IEEE (2012)
6. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: 2007 IEEE 11th International Conference on Computer Vision. pp. 1–8. IEEE (2007)
7. Fernando, B., Tuytelaars, T.: Mining multiple queries for image retrieval: On-the-fly learning of an object-specific mid-level representation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2544–2551 (2013)
8. Frowd, C., Bruce, V., McIntyre, A., Hancock, P.: The relative importance of external and internal features of facial composites. British journal of psychology **98**(1), 61–77 (2007)
9. Frowd, C.D., Erickson, W.B., Lampinen, J.M., Skelton, F.C., McIntyre, A.H., Hancock, P.J.: A decade of evolving composites: regression-and meta-analysis. Journal of Forensic Practice **17**(4), 319–334 (2015)
10. Galoogahi, H.K., Sim, T.: Inter-modality face sketch recognition. In: 2012 IEEE International Conference on Multimedia and Expo. pp. 224–229. IEEE (2012)
11. Hsiao, K., Calder, J., et al.: Pareto-depth for multiple-query image retrieval. IEEE transactions on image processing: a publication of the IEEE Signal Processing Society **24**(2), 583–594 (2015)
12. Hu, W., Hu, H.: Fine tuning dual streams deep network with multi-scale pyramid decision for heterogeneous face recognition. Neural Processing Letters **50**(2), 1465–1483 (2019)
13. Hu, Z., Liu, X., Wang, X., Cheung, Y.m., Wang, N., Chen, Y.: Triplet fusion network hashing for unpaired cross-modal retrieval. In: Proceedings of the 2019 on International Conference on Multimedia Retrieval. pp. 141–149. ACM (2019)
14. Huang, P.Y., Chang, X., Hauptmann, A.G., et al.: Improving what cross-modal retrieval models learn through object-oriented inter-and intra-modal attention networks. In: Proceedings of the 2019 on International Conference on Multimedia Retrieval. pp. 244–252. ACM (2019)
15. Jiang, J., Yu, Y., Wang, Z., Liu, X., Ma, J.: Graph-regularized locality-constrained joint dictionary and residual learning for face sketch synthesis. IEEE Transactions on Image Processing **28**(2), 628–641 (2018)
16. Jiao, L., Zhang, S., Li, L., Liu, F., Ma, W.: A modified convolutional neural network for face sketch synthesis. Pattern Recognition **76**, 125–136 (2018)
17. Kennedy, L.S., Naaman, M.: Generating diverse and representative image search results for landmarks. In: Proceedings of the 17th international conference on World Wide Web. pp. 297–306. ACM (2008)
18. Klare, B., Li, Z., Jain, A.K.: Matching forensic sketches to mug shot photos. IEEE transactions on pattern analysis and machine intelligence **33**(3), 639–646 (2010)
19. Lee, S.G., Vu, Q.P.: Simultaneous solutions of sylvester equations and idempotent matrices separating the joint spectrum. Linear Algebra and its Applications **435**(9), 2097–2109 (2011)

20. Lin, Z., Ding, G., Hu, M., Wang, J.: Semantics-preserving hashing for cross-view retrieval. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3864–3872 (2015)
21. Mohedano, E., McGuinness, K., O'Connor, N.E., Salvador, A., Marques, F., Giro-i Nieto, X.: Bags of local convolutional features for scalable instance search. In: Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval. pp. 327–331. ACM (2016)
22. Otto, C., Springstein, M., Anand, A., Ewerth, R.: Understanding, categorizing and predicting semantic image-text relations. In: Proceedings of the 2019 on International Conference on Multimedia Retrieval. pp. 168–176. ACM (2019)
23. Ouyang, S., Hospedales, T.M., Song, Y.Z., Li, X.: Forgetmenot: Memory-aware forensic facial sketch matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5571–5579 (2016)
24. Roy, H., Bhattacharjee, D.: Face sketch-photo matching using the local gradient fuzzy pattern. IEEE Intelligent Systems **31**(3), 30–39 (2016)
25. Sharma, A., Jacobs, D.W.: Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch. In: CVPR 2011. pp. 593–600. IEEE (2011)
26. Song, J., He, T., Gao, L., Xu, X., Shen, H.T.: Deep region hashing for generic instance search from images. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
27. Taghizadeh, M., Chalechale, A.: A novel method for multiple-query image retrieval. In: 2015 Signal Processing and Intelligent Systems Conference (SPIS). pp. 63–66. IEEE (2015)
28. Tang, X., Wang, X.: Face sketch recognition. IEEE Transactions on Circuits and Systems for Video Technology **14**(1), 50–57 (2004)
29. Uhl, R.G., da Vitoria Lobo, N.: A framework for recognizing a facial image from a police sketch. In: Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 586–593. IEEE (1996)
30. Wang, N., Tao, D., Gao, X., Li, X., Li, J.: A comprehensive survey to face hallucination. International journal of computer vision **106**(1), 9–30 (2014)
31. Wang, X., Tang, X.: Face photo-sketch synthesis and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(11), 1955–1967 (2008)
32. Wang, Y., Lin, X., Wu, L., Zhang, W.: Effective multi-query expansions: Collaborative deep networks for robust landmark retrieval. IEEE Transactions on Image Processing **26**(3), 1393–1404 (2017)
33. Yu, T., Wu, Y., Bhattacharjee, S., Yuan, J.: Efficient object instance search using fuzzy objects matching. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
34. Zhu, L., Shen, J., Jin, H., Xie, L., Zheng, R.: Landmark classification with hierarchical multi-modal exemplar feature. IEEE Transactions on Multimedia **17**(7), 981–993 (2015)