This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2944371, IEEE Transactions on Mobile Computing

A Cloud-guided Feature Extraction Approach for Image Retrieval in Mobile Edge Computing

Shangguang Wang, *Senior Member, IEEE,* Chuntao Ding, Ning Zhang, *Member, IEEE,* Xiulong Liu, Ao Zhou, *Member, IEEE,* Jiannong Cao, *Fellow, IEEE,* and Xuemin (Sherman) Shen, *Fellow, IEEE*

Abstract—Mobile Edge Computing (MEC) can facilitate various important image retrieval applications for mobile users by offloading partial computation tasks from resource-limited mobile devices to edge servers. However, existing related works suffer from two major limitations. (i) *High network bandwidth cost*: they need to extract numerous features from the image and upload these feature data to the cloud server. (ii) *Low retrieval accuracy*: they separate the feature extraction processes from the image data set in the cloud server, thus unable to provide effective features for accurate image retrieval. In this paper, we propose a cloud-guided feature extraction approach for mobile image retrieval. In the proposed approach, the cloud server first leverages the relationships among labeled images in the data set to learn a projection matrix $\mathbf{P}$. Then, it uses the matrix $\mathbf{P}$ to extract discriminative features from the image data set and form a low-dimensional feature data set. Follow that, the cloud server sends the matrix $\mathbf{P}$ to the edge server and uses it to multiply the image $\mathbf{x}$. The result $\mathbf{P}^T\mathbf{x}$, *i.e.*, image features, is uploaded to the cloud server to find the label of the image with the most similar multiplying result. The label is regarded as the retrieval result and returned to the mobile user. In the cloud-guided feature extraction approach, the matrix $\mathbf{P}$ can extract a small number of effective image features, which not only reduces network traffic but also improves retrieval accuracy. We have implemented a prototype system to validate the proposed approach and evaluate its performance by conducting extensive experiments using a real MEC environment and data set. The experimental results show that the proposed approach reduces the network traffic by nearly 93% and improves the retrieval accuracy by nearly 6.9% compared with the state-of-the-art image retrieval approaches in MEC.

Index Terms—Mobile Edge Computing, cloud-guided, feature extraction, image retrieval, edge servers.

1 Introduction

1.1 Motivation & Problem Statement

With the growing popularity of mobile devices, image retrieval approaches can facilitate various promising applications, *e.g.*, object identification for visually impaired individuals, and facial recognition for authentication [1]–[3]. The most popular solution is based on mobile cloud computing [4]–[6], *i.e.*, a mobile user uploads raw image data (or pre-processed data) to cloud servers, and then gets retrieval results from the cloud servers. However, directly uploading image-related data to cloud servers can incur a long network transmission delay. We use Mobile Edge Computing (MEC) [7]–[10] to solve the network transmission delay problem. This is because mobile users can launch image retrieval requests and get retrieval results from edge servers, which are closer to users than cloud servers. Moreover, MEC and image retrieval can mitigate many challenges of the Internet of Things (IoT), e-healthcare and autonomous car under the existing network and 5G environment. For example, various IoT devices can upload corresponding IoT data [11], [12] to edge servers to reduce response time [13]. MEC can enable e-healthcare to help patients access different healthcare assistance quickly [14]. In the autonomous car application, MEC can help obstacle detection systems quickly detect obstacles [15]. In these applications, the primary role of edge servers is to reduce the stress on the core network and transmission delay by pre-processing the uploaded data.

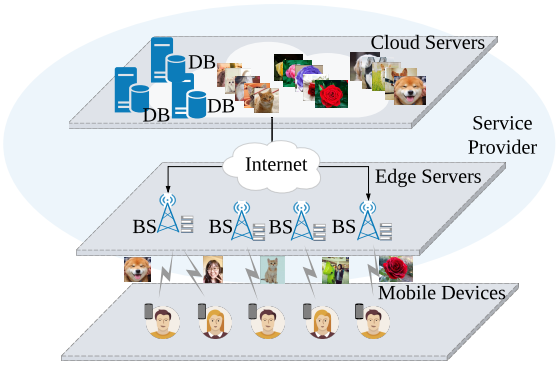


Fig. 1. System architecture of image retrieval in mobile edge computing.

However, most existing related MEC-based image retrieval approaches need to extract numerous features from

- *Shangguang Wang and Ao Zhou are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. E-mail: sgwang@bupt.edu.cn; aozhou@bupt.edu.cn.*
- *Chuntao Ding is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China and the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. E-mail: ctding@bupt.edu.cn.*
- *Xiulong Liu and Jiannong Cao are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. E-mail: xiulongliudut@gmail.com; csjcao@comp.polyu.edu.hk.*
- *Ning Zhang is with the Department of Computing Sciences, Texas A&M University-corpus Christi, Corpus Christi, USA. E-mail: ning.zhang@tamucc.edu.*
- *Xuemin (Sherman) Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada. E-mail: sshen@uwaterloo.ca.*
  Correspondence author: Xiulong Liu

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2944371, IEEE Transactions on Mobile Computing

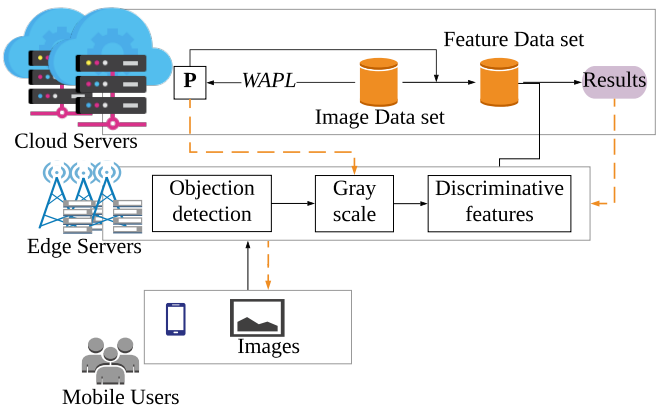


Fig. 2. The cloud-guided feature extraction approach for image retrieval.

the image since they aim to preserve its intrinsic structure. Hence, a large amount of feature data needs to be uploaded from edge servers to cloud servers. In addition, their feature extraction processes are isolated from image data set stored on the cloud servers. Thus, they cannot extract effective discriminative features and result in low retrieval accuracy.

In this paper, we study the problem of image retrieval in the MEC context, which is described as follows. As illustrated in Fig. 1, the system architecture of MEC consists of three layers of components: mobile devices (users), edge servers, and cloud servers. Mobile devices communicate with edge servers through LTE or WiFi and the edge servers connect to cloud servers via the Internet backbone. A large amount of labeled image data is stored on cloud servers. From the perspective of mobile users, edge servers and cloud servers are together regarded as a service provider. A mobile user uploads an image to the service provider to launch an image retrieval request. Then, the service provider processes and returns the label information of the most similar image to the mobile user.

1.2 Proposed Approach

In this paper, we propose a cloud-guided feature extraction approach for image retrieval in MEC, as shown in Fig. 2. We first, propose a Weight-Adaptive Projection matrix Learning algorithm (WAPL) to learn a projection matrix $\mathbf{P}$, which is used to extract discriminative features from the image data set on cloud servers to generate a low-dimensional feature data set. That is, we use the matrix $\mathbf{P}$ to multiply each image data in the image data set. The multiplying result can be interpreted as the discriminative features of the corresponding image. Then, cloud servers send the matrix $\mathbf{P}$ to edge servers. When receiving the image data, edge servers first pre-process it, such as objection detection and gray scale. Then, the edge servers use the matrix $\mathbf{P}$ to multiply the pre-processed image $\mathbf{x}$. The result $\mathbf{P}^T\mathbf{x}$, *i.e.*, feature data, is uploaded to cloud servers to find the label of an image with the most similar multiplying result. The label is regarded as the retrieval result and returned to mobile users. Since the matrix $\mathbf{P}$ can extract effective discriminative features from the image, edge servers just upload a small amount of feature data to cloud servers. Compared with existing MEC-based image retrieval approaches, our approach has less network traffic and faster responses. In addition, our approach can achieve higher retrieval accuracy.

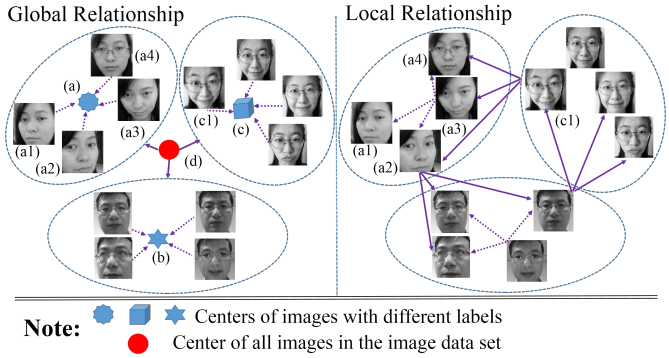


Fig. 3. Illustration of global and local relationships. (a1), (a2), (a3), (a4), (c1) are indexes of images, and (a), (b), (c) and (d) are indexes of centers of images. The global relationship includes the relationship between the image and the center of images with the same label and the relationship between the image centers and the center of all images. The local relationship includes the relationship between images with the same label and the relationship between images with different labels.

1.3 Challenges and Proposed Solutions

The first challenge is to guarantee that the projection matrix $\mathbf{P}$ *can extract effective discriminative features.* Some related MEC-based image retrieval approaches use the Local Binary Patterns (LBP) algorithm [16] to extract features. Since the LBP algorithm aims to preserve the intrinsic structure of the image, it needs to extract numerous features. In addition, a number of algorithms have been proposed for learning the projection matrix $\mathbf{P}$ to extract discriminative features. However, most of them either consider partial relationships (*i.e.*, global or local relationship) of the image data set or assign the same weights to global and local relationships (as illustrated in Fig. 3). Different relationships are equally treated, which is not reasonable for most image data sets. Since the importance of different relationships can be quite different, their weights need to be carefully decided. Therefore, we propose a WAPL algorithm where global and local relationships are divided into four types of dissimilarities. Moreover, the WAPL algorithm introduces trade-off parameters α, β and γ to control the weight of these dissimilarities. Thus, the learned matrix $\mathbf{P}$ can extract effective discriminative features.

The second challenge is to reduce manpower cost involved in determining the dimension of the feature data set. The matrix $\mathbf{P}$ is used to extract discriminative features from the image data set in cloud servers and the image in edge servers. Hence, it is necessary to determine the optimal number of discriminative features (*i.e.*, the dimension of the feature data set). However, most existing algorithms empirically estimate the optimal number of discriminative features, which may require a lot of manpower costs. To circumvent this problem, we theoretically study the relationship between the optimal number of discriminative features, the retrieval accuracy and the number of eigenvalues. Finally, we prove that the optimal number of discriminative features can be evaluated in terms of the number of positive eigenvalues. Thus, considerable manpower costs can be saved by determining the dimension of the feature data set in terms of the number of positive eigenvalues.

The third challenge is to meet the different requirements of users. In real-world scenarios, users have different requirements for accuracy and response time. For example, in an

authentication system, users pay more attention to retrieval accuracy compared with response time. However, in an autonomous driving system, obstacle detection needs to the real-time response. Therefore, it is necessary to design different interaction strategies between edge servers and cloud servers since both retrieval accuracy and response time depend on these strategies. Based on the results of the challenge 2, we develop different interaction strategies between edge servers and cloud servers. Thus, we can meet the different requirements of users.

1.4 Novelty and Advantage over Prior Art

The technical novelty of this paper is to propose a cloud-guided feature extraction approach. The technical depth of this paper is to learn a projection matrix, automatically determine the dimension of the feature data set, and meet various requirements of users. Compared with the state-of-the-art image retrieval approaches in MEC context, the key advantages of the proposed approach are two-fold: (i) Experimental results reveal that the proposed approach reduces the network traffic by 93%. (ii) The image retrieval accuracy is improved by 6.9%.

The remainder of this paper is organized as follows. Section 2 reviews the related work. The proposed MEC-based image retrieval approach is presented in Section 3. Section 4 introduces a novel projection matrix algorithm. The interaction strategies between edge servers and cloud servers are introduced in Section 5. In Section 6, we implement a prototype system to evaluate the performance of the approach. Section 7 discusses the proposed approach. Section 8 concludes this paper and outlines future work.

2 Related Work

2.1 Image Retrieval

For decades, image retrieval [17], [18] has been a hot research topic in the computer vision, with the goal of retrieving labels of similar images from data sets. In the following, we will discuss two main procedures in image retrieval systems, namely feature extraction and feature matching.

Feature extraction aims to extract features from the original high-dimensional data sets. In general, it consists of two steps. The first step is to learn the projection matrix. The second step is to use the projection matrix to extract features from the original image data set and form a low-dimensional feature data set. Local Binary Pattern (LBP) [16] and Principle Component Analysis (PCA) [19] are two classic feature extraction algorithms. LBP extracts features to preserve the intrinsic structure of the image. PCA extracts features to preserve the global information of the image. However, since they do not utilize the label information of the image, it is difficult to extract discriminative features useful for image retrieval. To address this problem, a number of algorithms for using the label information to learn the projection matrix are proposed, such as [20]–[23]. For example, Linear Discriminant Analysis (LDA) [20] aims to preserve the global relationship of the image data. Marginal Fisher Analysis (MFA) [21] focuses on the local relationship of the image data. Joint Global and Local-structure Discriminant Analysis (JGLDA) [22] considers both global and local structures. However, JGLDA treats the importance of both structures equally when dealing with different data sets. In practice, the importance of different relationships when extracting features on different data sets is different. In addition, above algorithms are empirical to estimate the dimension of the extracted feature data set. The inability to automatically determine the dimension of the extracted feature data set affects their applications because tuning it requires considerable manpower costs.

Feature matching aims to design effective classifiers to recognize different images. There are multi-class classifiers, such as the nearest neighbor classifier [24] and support vector machine [25]. Feature matching is the most time-consuming procedure in a real image retrieval system because the image to be retrieved needs to match all the images stored in the image data set, and the images stored in the image data set are high-dimensional.

2.2 Mobile Edge Computing

Mobile Edge Computing (MEC) [26], [27] has recently become a new computing paradigm with proximate access and is a promising complement to centralized Mobile Cloud Computing (MCC). In the MEC paradigm, a number of small scales edge servers are placed at the edge of the network. Mobile users can connect these edge servers via LTE or WiFi connection. The main idea of MEC is to deploy edge servers on the edge of the network close to the user so that users can use the computing, storage and other resources provided by the edge servers. Compared with MCC, the network traffic and network transmission time of MEC architecture can be significantly reduced because the edge servers are closer to mobile users.

A lot of research has been carried out on MEC [28]–[37]. For example, Soyata *et al.* [9] proposed a mobile-cloudlet-cloud architecture designed to perform task load between cloud servers to minimize response time. Hu *et al.* [10] proposed a face identification and resolution scheme based on fog computing, which can reduce network traffic by offloading partial process of the image data on fog nodes. Liu *et al.* [28] proposed a food recognition system based edge computing architecture that pre-processes the captured food image on the mobile devices before uploading it to cloud servers. However, they need to upload a large amount of data from edge servers to cloud servers because they extract numerous features from the image to preserve its intrinsic structure. In addition, they cannot extract effective discriminative features because their feature extraction processes are isolated from image data set stored on cloud servers.

3 The Cloud-guided Feature Extraction Approach

In this section, we present the architecture of our MEC-based image retrieval system and describe the detailed design of the cloud-guided feature extraction approach.

The proposed system architecture consists of three layers of components: mobile users (devices), *e.g.*, smartphones; edge servers, *e.g.*, micro servers; and cloud servers, *e.g.*, Alibaba cloud servers. In general, a large number of image data sets are stored on cloud servers. The image data sets are usually high-dimensional, which contain a large number of redundant features. These features not only impair retrieval accuracy but also incur long feature matching time.

As shown in Fig. 2, we first propose the WAPL algorithm. Then, we perform the WAPL algorithm to learn the projection matrix $\mathbf{P}$ on the image data set of cloud servers. Follow that, we use the matrix $\mathbf{P}$ to extract discriminative features from the image data set and form a low-dimensional feature data set, *i.e.*, $\mathbf{P}^T\mathbf{X}$. The result $\mathbf{P}^T\mathbf{X}$ satisfies that, if images have the same label, their features are compact; otherwise, their features are separable. In other words, if two images $\mathbf{x}_i$ and $\mathbf{x}_j$ have the same label, the results $\mathbf{P}^T\mathbf{x}_i$ and $\mathbf{P}^T\mathbf{x}_j$ will be quite similar; otherwise, the results will be significantly different. Then, cloud servers send the matrix $\mathbf{P}$ to edge servers. When mobile users use mobile devices to capture images and launch image retrieval requests, they first upload image data to edge servers via LTE or WiFi. When receiving the image data, edge servers first pre-process it. For example, edge servers perform object detection algorithm to extract object region, remove unrelated regions [38], and resize the object region. In addition, edge servers convert the object region to gray scale image [10]. Note that, the pre-processed image should be a particular size. That is, the number of rows of $\mathbf{P}$ is the same as the number of rows of the pre-processed image $\mathbf{x}$. Then, edge servers use the matrix $\mathbf{P}$ to extract discriminative features from the pre-processed image $\mathbf{x}$, *i.e.*, $\mathbf{P}^T\mathbf{x}$, and upload $\mathbf{P}^T\mathbf{x}$ to cloud servers. When receiving the image feature data $\mathbf{P}^T\mathbf{x}$, cloud servers perform the feature matching algorithm (*e.g.*, the nearest neighbor classifier [24]) to find the most similar images in data sets. Note that, two images are similar when the Euclidean distance of their feature data is small. Finally, cloud servers send back the labels of the most similar images to edge servers and the edge servers send back these labels as retrieval results to mobile users. In addition, if the image is not in the data set, the retrieved results are also the labels of the image on the cloud server that is most similar to the image. That is, the process of retrieving a new image is similar to retrieving an image in a data set. Hence, retrieving new images and retrieving images from the data set all benefit from the proposed approach. Furthermore, the proposed approach can be applied to identify and retrieve any image because the proposed approach is general.

Our approach has three advantages compared with existing MEC-based image retrieval approaches. First, our approach consumes less core network bandwidth because it uses matrix $\mathbf{P}$ to extract discriminative features from the image. Thus, edge servers only need to upload less feature data to cloud servers and consume less core network bandwidth. Second, our approach can provide faster responses. The response time mainly consists of network transmission time and feature matching time. Edge servers consume less network transmission time since they only upload less feature data to cloud servers. In addition, the feature matching operation takes less time because it is performed in the low-dimensional feature data set space. Hence, our approach can achieve a lower response time. Third, our approach can provide higher retrieval accuracy because we perform the WAPL algorithm on the image data sets of cloud servers to obtain matrix $\mathbf{P}$. In other words, image data sets stored on cloud servers guide the extraction of discriminative features from the image data on edge servers. Therefore, using matrix $\mathbf{P}$ can extract effective discriminative features and achieve higher retrieval accuracy.

TABLE 1
Frequently Used Notations

Symbol	Descriptions
$\mathbf{X}$	an image data set, where $\mathbf{X}=\{\mathbf{x}_i\}_{i=1}^N$
$\mathbf{x}_i$	i-th image
N	number of images
d	dimensionality of the images
$\mathbf{Y}$	corresponding label matrix, where $\mathbf{Y}=\{y_i\}_{i=1}^C$
C	number of classes
r	dimension of the feature data set space
$\mathbf{P}$	projection matrix, where $\mathbf{P}^T\mathbf{P}=\mathbf{I}$
$\mathbf{I}$	identity matrix
μ^m	mean of the images in class m
N_m	number of images in class m
μ	mean of all the images
$\mathbf{x}_i^m$	i-th image in class m
f_{gw}	global intra-class dissimilarity
f_{gb}	global inter-class dissimilarity
f_{lw}	local intra-class dissimilarity
f_{lb}	local inter-class dissimilarity
$\mathbf{L}_{lw}$	Laplacian matrix, where $\mathbf{L}_{lw}=\mathbf{D}^{lw}-\mathbf{W}^{lw}$
$\mathbf{L}_{lb}$	Laplacian matrix, where $\mathbf{L}_{lb}=\mathbf{D}^{lb}-\mathbf{W}^{lb}$
$\mathbf{W}^{lw}$, $\mathbf{W}^{lb}$	symmetric similarity matrices
$\mathbf{D}^{lw}$, $\mathbf{D}^{lb}$	diagonal matrices, *i.e.*, $D_{ii}^{lw}=\sum_j W_{ij}^{lw}$, $D_{ii}^{lb}=\sum_j W_{ij}^{lb}$
$\mathbf{S}_w$, $\mathbf{S}_b$	intra-class/inter-class scatter matrices

4 Projection Matrix Learning Algorithm

So far, the unclear issue of the proposed approach is to learn the projection matrix $\mathbf{P}$, which is important because the matrix $\mathbf{P}$ is used to extract discriminative features from the image data set on cloud servers and the image on edge servers. Moreover, the matrix $\mathbf{P}$ determines whether the extracted discriminative features are effective. In this section, we propose the WAPL to learn the matrix $\mathbf{P}$. Frequently used notations are summarized in Table 1.

To ensure that the matrix $\mathbf{P}$ can extract effective discriminative features, the WAPL algorithm should contain both global and local relationships. This is because global and local relationships are beneficial to make the matrix $\mathbf{P}$ extract effective discriminative features. To this end, the WAPL algorithm first divides the traditional global and local relationships into four types of dissimilarities: global intra-class, global inter-class, local intra-class, and local inter-class dissimilarities. Fig. 3 shows an example of these dissimilarities. These four types of dissimilarities are more granular than global and local relationships. Motivated by [20], [21], we first give the quantification of these four types of dissimilarities. The global intra-class dissimilarity f_{gw} indicates the relationship between image $\mathbf{x}_i^m$ and μ^m, which can be quantified as:

$$f_{gw} = \sum_{m=1}^{C}\sum_{i=1}^{N_m}\mathbf{P}^T(\mathbf{x}_i^m-\mu^m)(\mathbf{x}_i^m-\mu^m)^T\mathbf{P} \quad . \tag{1}$$

The global inter-class dissimilarity f_{gb} indicates the relationship between μ^m and μ, which can be quantified as:

$$f_{gb} = \sum_{m=1}^{C} N_m\mathbf{P}^T(\mu^m-\mu)(\mu^m-\mu)^T\mathbf{P} \quad . \tag{2}$$

The local intra-class dissimilarity f_{lw} indicates the pairwise relationship between the images with the same label, which can be quantified as:

$$f_{lw}=\sum_{ij}||\mathbf{P}^T\mathbf{x}_i-\mathbf{P}^T\mathbf{x}_j||^2W_{ij}^{lw} \quad , \tag{3}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2944371, IEEE Transactions on Mobile Computing

$$W_{ij}^{lw}=\begin{cases} e^{-\frac{||\mathbf{x}_i-\mathbf{x}_j||^2}{t_1}}, i\in NS_{k_1}^{w}(j) \text{ or } j\in NS_{k_1}^{w}(i) \\ 0, \text{ otherwise} \end{cases}, \quad (4)$$

where $NS_{k_1}^{w}(i)$ denotes the index set of the k_1 nearest neighbors of image $\mathbf{x}_i$ with the same label. t_1 is a constant parameter, which controls how rapidly the W_{ij}^{lw} falls off with the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. In general, the constant t_1 is equal to the square of the mean of the distance between $\mathbf{x}_i$ and its k_1 nearest neighbors [39].

The local inter-class dissimilarity f_{lb} indicates the pairwise relationship between the images with different labels, which can be quantified as:

$$f_{lb}=\sum_{ij}||\mathbf{P}^T\mathbf{x}_i-\mathbf{P}^T\mathbf{x}_j||^2W_{ij}^{lb}, \quad (5)$$

$$W_{ij}^{lb}=\begin{cases} e^{-\frac{||\mathbf{x}_i-\mathbf{x}_j||^2}{t_2}}, i\in NS_{k_2}^{b}(j) \text{ or } j\in NS_{k_2}^{b}(i) \\ 0, \text{ otherwise} \end{cases}, \quad (6)$$

where $NS_{k_2}^{b}(i)$ denotes the index set of k_2 nearest neighbors of image $\mathbf{x}_i$ with different labels. The constant t_2 is equal to the square of the mean of the distance between $\mathbf{x}_i$ and its k_2 nearest neighbors.

Combined with Fig. 3, we explain Eqs. (1-6) as follows. Eq. (1) reflects the relationship between images and the center of images with the same label, *e.g.*, the relationship between (a1), (a2), (a3), (a4) and (a). Eq. (2) reflects the relationship between the centers of images with the same label and the center of all images, *e.g.*, the relationship between (a), (b), (c) and (d). Eq. (3) reflects the relationship between images with the same label, *e.g.*, the relationship between (a1), (a2), (a4) and (a3). Eq. (5) reflects the relationship between images with different labels, *e.g.*, the relationship between (a2), (a3), (a4) and (c1). Eq. (4) and Eq. (6) indicate adjacency matrices. Eq. (4) indicates the intra-class adjacency matrix, which aims to preserve the structure of images with the same label. Eq. (6) indicates the inter-class adjacency matrix, which aims to preserve the structure of images with different labels.

To improve the retrieval accuracy, it is necessary to minimize f_{gw} and f_{lw} while maximizing f_{gb} and f_{lb}. However, for most data sets, it is unreasonable to simply integrate these dissimilarities. This is because the importance of different types of dissimilarities can be quite different when the projection matrix learning algorithm processes different data sets. Therefore, we introduce three trade-off parameters α, β and γ to control their weights. However, most existing projection matrix learning algorithms use the Fisher criterion [21] to formalize the objective function. Although these algorithms incorporate all types of dissimilarities, they are difficult to reasonably control the weight of each type of dissimilarity. Motivated by [40], the objective function is defined as follows:

$$\begin{aligned} &\max_{\mathbf{P}}[\gamma\beta f_{gb}+\gamma(1-\beta)f_{lb}]-[\alpha(1-\gamma)f_{gw} \\ &\quad +(1-\gamma)(1-\alpha)f_{lw}] \\ &s.t.\ \mathbf{P}^T\mathbf{P}=\mathbf{I} \end{aligned}, \quad (7)$$

where $\alpha,\beta,\gamma\in[0,1]$ are trade-off parameters that reflect the importance between f_{gw} and f_{lw}, the importance between g_{gb} and g_{lb}, and the importance between $\alpha f_{gw}+(1-\alpha)g_{lw}$ and $\beta f_{gb}+(1-\beta)f_{lb}$, respectively. When $\alpha=\beta=\gamma=0$, Eq. (7) aims to minimize f_{lw}. When $\alpha=\beta=\gamma=1$, Eq. (7) aims to maximize f_{gb}. When $\alpha,\beta,\gamma\in(0,1)$, f_{gb} and f_{lb} can be maximized while f_{gw} and f_{lw} can be minimized.

In doing so, the importance of all types of dissimilarities can be controlled according to the requirements in different image data sets. For brevity, Eq. (7) can be rewritten as:

$$\mathbf{P}^*=\arg\max_{\mathbf{P}} tr(\mathbf{P}^T\mathbf{H}\mathbf{P}) \ \ s.t.\ \mathbf{P}^T\mathbf{P}=\mathbf{I}, \quad (8)$$

where

$$\begin{aligned} \mathbf{H}=&\gamma\beta\mathbf{S}_b-(1-\gamma)\alpha\mathbf{S}_w+\mathbf{X}[2\gamma(1-\beta)\mathbf{L}_{lb} \\ &-2(1-\gamma)(1-\alpha)\mathbf{L}_{lw}]\mathbf{X}^T \end{aligned}. \quad (9)$$

Eq. (8) ensures that the matrix $\mathbf{P}$ can extract effective discriminative features because the WAPL algorithm includes all types of dissimilarities. Moreover, the WAPL algorithm controls the importance of these dissimilarities by using trade-off parameters.

In addition, the WAPL algorithm runs on cloud servers, the optimal dimension of the feature data set should be automatically estimated to avoid a lot of manpower costs. Since different data sets correspond to different optimal dimensions, it is difficult to empirically estimate the optimal dimensions of all data sets. To address this problem, we first study the properties of $\mathbf{H}$. Eq. (1) and Eq. (2) exhibit that $\mathbf{x}_i^m-\mu^m$ and $\mu^m-\mu$ are real matrices. Thus, based on [41], $\mathbf{S}_w^T$ can be written as:

$$\begin{aligned} \mathbf{S}_w^T&=\sum_{m=1}^{C}\sum_{i=1}^{N_m}\{(\mathbf{x}_i^m-\mu^m)(\mathbf{x}_i^m-\mu^m)^T\}^T \\ &=\sum_{m=1}^{C}\sum_{i=1}^{N_m}\{(\mathbf{x}_i^m-\mu^m)^T\}^T\{(\mathbf{x}_i^m-\mu^m)^T\}. \\ &=\sum_{m=1}^{C}\sum_{i=1}^{N_m}(\mathbf{x}_i^m-\mu^m)(\mathbf{x}_i^m-\mu^m)^T=\mathbf{S}_w \end{aligned} \quad (10)$$

Similarly, $\mathbf{S}_b^T$ can be written as:

$$\begin{aligned} \mathbf{S}_b^T&=\sum_{m=1}^{C}N_m\{(\mu^m-\mu)(\mu^m-\mu)^T\}^T \\ &=\sum_{m=1}^{C}N_m\{(\mu^m-\mu)^T\}^T\{(\mu^m-\mu)^T\}. \\ &=\sum_{m=1}^{C}N_m(\mu^m-\mu)(\mu^m-\mu)^T=\mathbf{S}_b \end{aligned} \quad (11)$$

Hence, $\mathbf{S}_w$ and $\mathbf{S}_b$ are real symmetric matrices. In addition, according to Eq. (4) and Eq. (6), $\mathbf{W}^{lw}$ and $\mathbf{W}^{lb}$ are symmetric matrices. Moreover, since the real diagonal matrix must be a real symmetric matrix according to [41], $\mathbf{D}^{lw}$ and $\mathbf{D}^{lb}$ are symmetric matrices. Thus, $\mathbf{L}_{lw}=\mathbf{D}^{lw}-\mathbf{W}^{lw}$ and $\mathbf{L}_{lb}=\mathbf{D}^{lb}-\mathbf{W}^{lb}$ are real symmetric matrices. Since $\mathbf{S}_b$, $\mathbf{S}_w$, $\mathbf{L}_{lb}$ and $\mathbf{L}_{lw}$ are real symmetric matrices, based on Eq. (9) and [41], $\mathbf{H}$ is a real symmetric matrix. In addition, it is also non-positive definite and the eigenvalues of $\mathbf{H}$ can be positive, zero, or negative. This motivates us to solve Eq. (8) by utilizing the relationships between the eigenvalues of $\mathbf{H}$, the eigenvectors of $\mathbf{H}$ and $\mathbf{H}$. According to [40], [41], we propose Theorem 1 in the following.

Theorem 1. *The solution $\mathbf{P}^*$ of the objective function in Eq. (8) is composed of eigenvectors $[\mathbf{p}_0,\cdots,\mathbf{p}_{r-1}]$ of $\mathbf{H}$ corresponding to the top r positive eigenvalues, where r is the number of positive eigenvalues of $\mathbf{H}$.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2944371, IEEE Transactions on Mobile Computing

Proof. The Lagrangian function of problem in Eq. (8) is:

$$\zeta(\mathbf{P}, \mathbf{\Lambda}) = tr(\mathbf{P}^T\mathbf{H}\mathbf{P}) - tr(\mathbf{\Lambda}(\mathbf{P}^T\mathbf{P} - \mathbf{I})) \ , \quad (12)$$

where $\mathbf{\Lambda} = [\lambda_1, \ldots, \lambda_n]$. By calculating its derivative with respect to $\mathbf{P}$ and setting it to zero, we have $\mathbf{H}\mathbf{p}_i = \lambda_i \mathbf{p}_i$. Thus, Eq. (8) can be rewritten as:

$$tr(\mathbf{P}^T\mathbf{H}\mathbf{P}) = \sum_{i=0}^{d-1} \mathbf{p}_i^T \mathbf{H}\mathbf{p}_i = \sum_{i=0}^{d-1} \mathbf{p}_i^T \lambda_i \mathbf{p}_i = \sum_{i=0}^{d-1} \lambda_i. \quad (13)$$

From Eq. (13), to maximize $tr(\mathbf{P}^T\mathbf{H}\mathbf{P})$, only the positive eigenvalues should be chosen since zero eigenvalues have no effect on $tr(\mathbf{P}^T\mathbf{H}\mathbf{P})$, and negative eigenvalues are harmful to $tr(\mathbf{P}^T\mathbf{H}\mathbf{P})$. The solution to Eq. (8) must be

$$\mathbf{P}^* = [\mathbf{p}_0, \cdots, \mathbf{p}_{r-1}]. \quad (14)$$

Hence, the statements in this theorem are proved. □

The optimal projection matrix $\mathbf{P}^*$ consists of eigenvectors corresponding to the top r positive eigenvalues based on Theorem 1. Fig. 4 depicts the relationship between accuracy and the eigenvectors corresponding to the eigenvalues. As analyzed in Theorem 1, the eigenvectors corresponding to the positive eigenvalues are advantageous for extracting discriminative features. The eigenvectors corresponding to the zero eigenvalues are useless for extracting discriminative features. The eigenvectors corresponding to the negative eigenvalues are detrimental to extracting discriminative features. Thus, as shown in Fig. 4, when the matrix $\mathbf{P}$ consists of eigenvectors corresponding to r positive eigenvalues, the WAPL algorithm can achieve the highest accuracy, *e.g.*, 95%. The accuracy of the WAPL algorithm tends to stabilize with the increase of the number of zero eigenvalues. It means the ability of the matrix $\mathbf{P}$ to extract discriminative features is invariable with the increase of the number of zero eigenvalues. The accuracy of the WAPL algorithm decreases with the increase of the number of negative eigenvalues, *i.e.*, the accuracy of the WAPL algorithm is lower than 95%. Hence, the value of r can be estimated, which is equals to the number of positive eigenvalues of $\mathbf{H}$. In other words, the optimal dimension of the feature data set space can be automatically estimated based on the number of positive eigenvalues rather than empirically. Therefore, the proposed approach can save a lot of manpower costs.

In addition, given N images of dimension d, the computational complexity of the WAPL algorithm is divided into three parts. First, the WAPL algorithm needs to compute the distance between μ^m and μ, the distance between $\mathbf{x}_i$ and μ^m, and the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. The computational complexity of the first part is $O(cdN)$. Second, the WAPL algorithm needs to construct adjacency graphs. The computational complexity of the second part is $O(dN^2)$. Third, the WAPL algorithm performs Eigen-decomposition on $\mathbf{H}$. The computational complexity of the third part is $O(d^3)$. In general, the dimensions of the image data set are relatively high. Thus, the computational complexity of the WAPL algorithm is $O(d^3)$.

5 Interaction Strategy

In real-world scenarios, users have different requirements in terms of retrieval accuracy and response time. We divide the requirements of users into three categories in terms of different scenarios. Scenario I: users pay more attention to retrieval accuracy. For example, in an authentication system, users can tolerate long response time (*e.g.*, within several seconds) but expect high retrieval accuracy. Scenario II: users pay more attention to response time. For example, in an autonomous driving system, obstacle detection needs the real-time response because users may not need to know exactly what the obstacles is. Scenario III: users pay attention to both retrieval accuracy and response time.

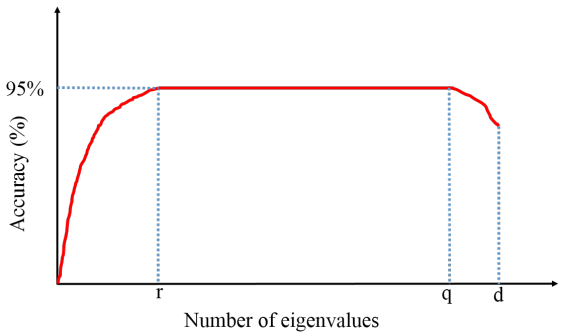


Fig. 4. The contour of Theorem 1. The eigenvalues have been sorted in reverse order. r, q, and d are indexes of eigenvalues. There are r positive eigenvalues, $q-r$ zero eigenvalues, and $d-q$ negative eigenvalues.

To meet the requirements of users in terms of retrieval accuracy and response time, different interaction strategies between edge servers and cloud servers are required. This is because both retrieval accuracy and response time depend on different network traffic (*i.e.*, the feature data) from edge servers to cloud servers. That is, when edge servers upload little discriminative feature data to cloud servers, the network transmission time and feature matching time can be reduced. However, little feature data incurs low accuracy. In contrast, when edge servers upload all discriminative feature data to cloud servers, users can achieve high accuracy. However, users have to tolerate long response time. In addition, Theorem 1 exhibits that the optimal dimension of the feature data set can be evaluated in terms of the number of positive eigenvalues. In other words, the optimal dimension of the feature data set is determined and equal to the dimension of the matrix $\mathbf{P}$. Hence, we can design different interaction strategies between edge servers and cloud servers according to the conclusion of Theorem 1.

Based on Theorem 1, when the matrix $\mathbf{P}$ consists of eigenvectors corresponding to all positive eigenvalues, it can extract all discriminative features and enable users to achieve the highest accuracy. Hence, for the Scenario I that pursues the highest accuracy, the matrix $\mathbf{P}$ needs to be composed of eigenvectors corresponding to all positive eigenvalues. That is, $\mathbf{P} = [\mathbf{p}_0, \cdots, \mathbf{p}_{r-1}]$, where r is the number of positive eigenvalues.

The matrix $\mathbf{P}$ containing eigenvectors corresponding to all positive eigenvalues is not suitable for scenarios that require a fast response. This is because the matrix $\mathbf{P}$ extracts numerous discriminative features. Uploading these feature data from edge servers to cloud servers results in a large amount of network traffic and long network transmission time. Moreover, the dimension of the feature data set is the dimension of $\mathbf{P}^T\mathbf{x}$. Performing the feature matching operation will also incur long matching time since the dimension of $\mathbf{P}^T\mathbf{x}$ is high. Hence, the matrix $\mathbf{P}$ is not suitable for scenarios that require fast responses. For Scenario II, since it requires fast responses and can tolerate low retrieval

accuracy, we can use a part of the matrix $\mathbf{P}$ in Scenario I as the projection matrix of Scenario II. To distinguish the matrix $\mathbf{P}$ of the Scenario I, we denote matrix $\mathbf{P}_2$ as the projection matrix in Scenario II. Thus, based on Theorem 1, we use a part of the matrix $\mathbf{P}$ as the projection matrix, then $\mathbf{P}_2 = \mathbf{P}(:, 1:z)$, where $z \in [1, r)$ is a positive integer and $\mathbf{P}(:, 1:z)$ refers to the first z columns of the matrix $\mathbf{P}$. Since Scenario II requires a fast response, we specify $z \ll r$. The matrix $\mathbf{P}_2$ extracts less discriminative features compared with the matrix $\mathbf{P}$. Although the matrix $\mathbf{P}_2$ results in lower retrieval accuracy, it can meet the user's requirements for a fast response.

Scenario III requires both retrieval accuracy and response time. We cannot use matrix $\mathbf{P}$ and $\mathbf{P}_2$ as the projection matrix in Scenario III because they only consider retrieval accuracy or response time. We denote matrix $\mathbf{P}_3$ as the projection matrix in Scenario III. Thus, we introduce a positive integer o, where $z \leq o \leq r$. Similar to Scenario II, we use a part of the matrix $\mathbf{P}$ as the projection matrix of Scenario III, then $\mathbf{P}_3 = \mathbf{P}(:, 1:o)$. The matrix $\mathbf{P}_3$ is a tradeoff between the matrix $\mathbf{P}$ and the matrix $\mathbf{P}_2$. The matrix $\mathbf{P}_3$ extracts less discriminative features than the matrix $\mathbf{P}$. Although matrix $\mathbf{P}_3$ results in lower retrieval accuracy, it can meet the user's requirements for a fast response. In addition, the matrix $\mathbf{P}_3$ extracts more discriminative features than the matrix $\mathbf{P}_2$. Although the matrix $\mathbf{P}_3$ results in longer responses, it can meet the user's requirements for high retrieval accuracy.

6 Performance Evaluation

In this section, we first evaluate the WAPL algorithm on three benchmark data sets. Then, we implement a prototype system to evaluate the proposed approach in a practical network environment with a real data set.

6.1 Experiment Setup

The experimental environment consists of a mobile device, three edge servers, and a cloud server.

- *Mobile Device:* A Huawei honor 8 smartphone is used as a mobile device. This smartphone is equipped with 4 Cortex A72 2.3 GHz and Android 7.0. It also has a 32 GB internal storage and 4 GB RAM. We develop an APP called "ImagCat" to capture images and upload them to the edge server and cloud server.
- *Edge Servers:* One of the edge servers is a computer equipped with Intel i5-4590@3.3 GHz CPU and 12 GB RAM. On the edge server, we use Java to invoke OpenCV libraries to pre-process images. In addition, we build a base station that is responsible for communicating with the mobile device and cloud server. Note that, the base station is next to the edge server. The base station is based on the Open Air Interface (OAI) [42]–[44], and consists of three components: radio-frequency signal generator, edge server A, and edge server B. The radio-frequency signal generator is equipped with USRP-B210. The edge server A is equipped with an Intel i7-6700@3.4 GHz CPU and 16 GB RAM for running the eNodeB. The radio-frequency signal generator and edge server A are connected via USB 3.0. The edge server B is equipped with Intel i5-6500@3.2 GHz CPU and 4 GB RAM for running Home Subscriber Service (HSS), Mobility Management (MME), Serving Gateway (SGW), and Packet data network Gateway (PGW) [44]–[46]. Edge servers A and B are next to each other and connected via the Local Area Network (LAN). The base station works on Band7 (uplink 2500 MHz-2570 MHz, downlink 2620 MHz-2690 MHz). The mobile device can connect to edge servers via LTE or WiFi. In the LTE situation, the upload link rate is 1000 KB/s and the download link rate is 1.36 MB/s. In the WiFi situation, we use a wireless router that connects to the campus network. The upload link rate and the download link rate are set to 9 MB/s. In this paper, we do not consider the coexistence of WiFi and LTE in the unlicensed band [47]–[49]. As the licensed band is very limited, making use of the unlicensed band will indeed significantly improve the wireless communication throughput. However, the coexistence of multiple wireless technologies that simultaneously use the unlicensed band will inevitably cause interference. We will leave the study on the coexistence of WiFi and LTE in the unlicensed band as our future work. The distance between the mobile device and the eNodeB and the distance between the mobile device and the AP are 0-10 meters.
- *Cloud Server:* Alibaba cloud server is equipped with 4 quad-core 2.5 GHz Intel Xeon E5-2682 v4 and 16 GB RAM as the cloud server. The cloud server runs Ubuntu 14.04.3 and implements the WAPL algorithm and feature matching algorithm by Python. Edge servers and cloud server are connected through the Internet backbone.

Fig. 5. The images cropped from Lab_face data set.

TABLE 2
Description of Benchmark Data Sets

Data set	#Images	#Features	#Classes
YaleB	2414	1024	38
UMIST	574	1024	20
USPS	9298	256	10
Lab_face	420	1024	21

6.2 Data Sets

We first evaluate the WAPL algorithm on extended Yale face database B (YaleB) [50], UMIST [51] and USPS handwritten digits (USPS) data sets [52]. Then, we collect a new data set Lab_face [53] and implement a prototype system to evaluate the proposed approach by using a real network environment. Table 2 lists the details of the benchmark data sets used in the experiment. In addition, Fig. 5 shows some examples of the Lab_face data set.

6.3 Comparison Algorithms and Approaches

6.3.1 Projection Matrix Learning Algorithms

We compare the WAPL algorithm with four state-of-the-art projection matrix learning algorithms: marginal Fisher

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2944371, IEEE Transactions on Mobile Computing

TABLE 3
Comparison of Image Retrieval Approaches

Approach	Edge server	WAPL	Edge server with $\mathbf{P}$
MCC_{simple}	No	No	No
MCC_{WAPL}	No	Yes	No
MEC_{simple}	Yes	No	No
MEC_{WAPL}	Yes	Yes	No
Our approach	Yes	Yes	Yes

analysis (MFA) [21], joint global and local-structure discriminant analysis (JGLDA) [22], double adjacency graphs-based discriminant neighborhood embedding (DAG-DNE) [40] and locality adaptive discriminant analysis (LADA) [23]. Algorithms are described in detail as follows.

- MFA [21] was introduced by Yan *et al.* in 2007, which learns the projection matrix by characterizing the intra-class compactness and inter-class separability.
- JGLDA [22] was introduced by Gao *et al.* in 2013, which learns the projection matrix by characterizing both the similarity and diversity of the image data.
- DAG-DNE [40] was introduced by Ding and Zhang in 2015, which learns the projection matrix by preserving the local pairwise relationship between images.
- LADA [23] was introduced by Li *et al.* in 2017, which learns the projection matrix by preserving the local pairwise relationship between images. In addition, it solves the problem of making assumptions about data distribution by linear discriminant analysis [20].

6.3.2 Related Image Retrieval Approaches

To evaluate the performance of the proposed approach, we compared it with other four image retrieval approaches. Approaches are described in detail as follows, and the main differences of them are given in Table 3.

- MCC_{simple}: in MCC_{simple} approach, a user first uses "ImagCat" to capture an image. Then, the user uploads the image data to the cloud server. When receiving the image data, the cloud server first pre-processes it. Then, the cloud server runs the LBP algorithm to extract features from the pre-processed image data and uses the matching algorithm to achieve results. Finally, the cloud server sends back the results to the user.
- MCC_{WAPL}: Different from MCC_{simple}, MCC_{WAPL} approach first uses the WAPL algorithm to learn the matrix $\mathbf{P}$. Then, it uses matrix $\mathbf{P}$ to extract features from the image data set. When receiving the image data, the cloud server first pre-processes it. Follow that, the cloud server uses the matrix $\mathbf{P}$ to extract discriminative features from the pre-processed image data. Finally, the matching algorithm is performed in the low-dimensional feature data set space.
- MEC_{simple}: Different from MCC_{simple}, in MEC_{simple} approach, the user first uploads the image data to the edge server. When receiving the image data, the edge server first pre-processes it. Then, the edge server uses the LBP algorithm to extract features from the pre-processed image data. Finally, the edge server uploads the extracted feature data to the cloud server.
- MEC_{WAPL}: Different from MEC_{simple}, MEC_{WAPL} approach first uses the WAPL algorithm to learn the matrix $\mathbf{P}$. Then, the matrix $\mathbf{P}$ is used to extract discriminative features from the image data set on the cloud server. When receiving the pre-processed image data, the cloud server uses the matrix $\mathbf{P}$ to further extract discriminative features from the pre-processed image data. Finally, the matching algorithm is performed in the low-dimensional feature data set space.
- Our approach: Different from MEC_{WAPL}, our approach uses the matrix $\mathbf{P}$ to extract discriminative features from the image data set on the cloud server. In addition, the cloud server sends the matrix $\mathbf{P}$ to the edge server to extract discriminative features from the image data. Thus, the edge server only uploads the extracted discriminative feature data to the cloud server.

TABLE 4
Image Retrieval Accuracy (% $\pm$ std)

Data Set	Algorithms	Results		
		$k=1$	$k=3$	$k=5$
YaleB	MFA	87.04±0.33	86.96±0.41	86.94±0.83
	JGLDA	87.08±0.58	86.56±0.83	86.96±0.58
	DAG-DNE	87.54±0.21	87.68±0.66	88.00±0.75
	LADA	88.52±0.25	88.52±0.25	88.52±0.25
	WAPL	**92.36±0.23**	**93.47±0.78**	**91.55±0.37**
UMIST	MFA	97.77±0.82	97.12±0.35	97.30±0.70
	JGLDA	97.65±0.67	97.89±0.32	97.12±0.66
	DAG-DNE	97.89±0.70	97.00±0.21	97.42±0.76
	LADA	97.31±0.43	97.31±0.43	97.31±0.43
	WAPL	**98.99±0.18**	**98.17±0.21**	**98.48±0.24**
USPS	MFA	85.84±0.43	88.41±0.44	89.77±0.97
	JGLDA	85.95±0.65	89.30±0.23	90.92±0.30
	DAG-DNE	92.38±0.64	92.23±0.86	92.51±0.14
	LADA	90.49±0.36	90.49±0.36	90.49±0.36
	WAPL	**95.89±0.46**	**95.24±0.15**	**96.68±0.31**

In this experiment, we compare the proposed approach with MCC_{simple}, MCC_{WAPL}, MEC_{simple} and MEC_{WAPL} approaches on Lab_face data set by using a real network environment. We choose five different sizes of images, which are related to the Lab_face data set, to evaluate the network traffic and response time. The image size order is Image1<Image2<Image3<Image4<Image5. For a fair comparison, we set the same value of the nearest-neighbor parameter k_1 and k_2 to construct adjacency graphs for all algorithms. Without prior knowledge, we set $k_1 = k_2$. For an easy display, we use k for k_1 and k_2. Finally, we use the nearest neighbor classifier to verify the extracted features.

6.4 Results of WAPL Algorithm

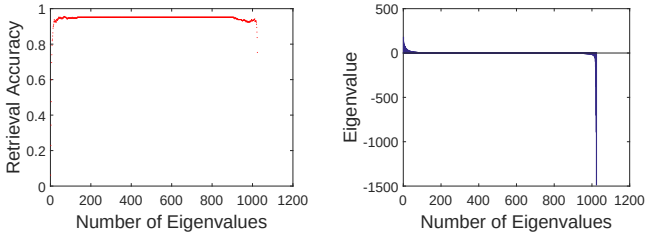
6.4.1 Retrieval Accuracy

In this section, we compare the WAPL algorithm with other state-of-the-art projection matrix learning algorithms. In the YaleB, UMIST and USPS data sets, 50% of images are randomly selected to form the training set, and the remaining images are used for testing.

Table 4 shows the experimental results. The WAPL algorithm achieves the highest retrieval accuracy in all image data sets. For example, on the YaleB data set, when $k = 3$, the accuracy of WAPL is 6.51% higher than the accuracy of MFA, 5.79% higher than the accuracy of DAG-DNE, and 4.95% higher than the accuracy of LADA. The reason is that WAPL incorporates all types of dissimilarities. However, MFA, DAG-DNE and LADA only contain partial types of dissimilarities. Losing partial types of dissimilarities impairs the ability of the projection matrix to extract discriminative features. Thus, they result in lower retrieval accuracy.

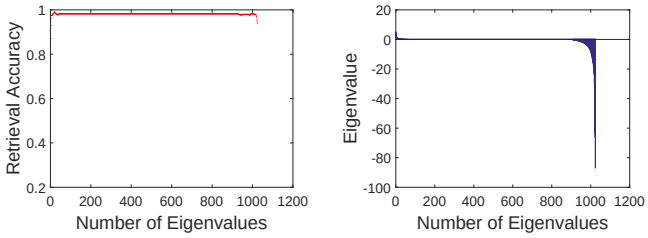
In addition, the accuracy of WAPL is 6.91% higher than the accuracy of JGLDA. The reason is that although JGLDA

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2944371, IEEE Transactions on Mobile Computing



(a) (b)

Fig. 6. Relationship between retrieval accuracy, eigenvalue, and the number of eigenvalues on the YaleB data set.



(a) (b)

Fig. 7. Relationship between retrieval accuracy, eigenvalue, and the number of eigenvalues on the UMIST data set.

incorporates all types of dissimilarities, it treats them equally. In practice, different dissimilarities contribute differently to learning the projection matrix. The equal treatment of each dissimilarity weakens the ability of the projection matrix to extract discriminative features.

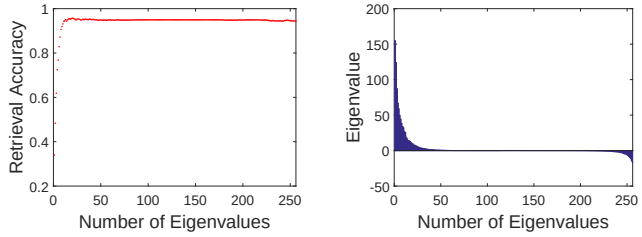
6.4.2 Relationship between Retrieval Accuracy and the Number of Eigenvalues

Figs. 6, 7, and 8 show that the retrieval accuracy of the WAPL algorithm increases rapidly as the number of positive eigenvalues increases. Then, the retrieval accuracy of the WAPL algorithm tends to stabilize as the number of zero eigenvalues increases. Finally, the retrieval accuracy of the WAPL algorithm decreases as the number of negative eigenvalues increases. It manifests that only eigenvectors corresponding to the positive eigenvalues contribute to extracting discriminative features. Moreover, the optimal dimension of the feature data set can be estimated in terms of the number of positive eigenvalues, which can save a lot of manpower costs in determining the optimal dimension of the feature data set. In addition, this discovery also helps us design different interaction strategies between edge servers and cloud servers to meet different requirements of users in terms of retrieval accuracy and response time. We will discuss it in detail in Section 6.5.4.

6.4.3 Parameters Analysis

The trade-off parameters α, β and γ can be tuned as follows. Each data set is randomly divided into a training set $\mathbf{X}_{Tr}$ and a test set $\mathbf{X}_{Te}$. The training set $\mathbf{X}_{Tr}$ is also randomly divided into a training set $\mathbf{X}_{Tr1}$ and a validation set $\mathbf{X}_{Va1}$. The training set $\mathbf{X}_{Tr1}$ is used to choose parameters, and the validation set $\mathbf{X}_{Va1}$ is used to validate parameters. α is evaluated by fixing β and γ, and varies from 0 to 1. β and γ are validated in the same way as α. Table 5 shows the corresponding parameter values when the WAPL algorithm achieves the highest retrieval accuracy on three data sets.

Table 5 shows that on different data sets, the corresponding parameter values are different when the WAPL



(a) (b)

Fig. 8. Relationship between retrieval accuracy, eigenvalue, and the number of eigenvalues on the USPS data set.

TABLE 5
Highest Accuracy and Corresponding Parameters

Data set	Retrieval Accuracy (%)	α	β	γ
YaleB	95.76	0.9	0.4	0
UMIST	99.69	1	0.5	0
USPS	96.54	0.5	0.5	0.1

algorithm achieves the highest retrieval accuracy. For example, on the YaleB data set, when $\alpha = 0.9$, $\beta = 0.4$ and $\gamma = 0$, the WAPL algorithm achieves the highest retrieval accuracy. Combined with Eq. (7), when $\alpha = 0.9$, $\beta = 0.4$ and $\gamma = 0$, Eq. (7) aims to minimize $0.9f_{gw} + 0.1f_{lw}$. This indicates that on the YaleB data set, effective discriminative features can be obtained by minimizing $0.9f_{gw} + 0.1f_{lw}$. In addition, f_{gw} is more important than f_{lw}. On the USPS data set, when $\alpha = 0.5$, $\beta = 0.5$ and $\gamma = 0.1$, the WAPL algorithm achieves the highest retrieval accuracy. When $\alpha = 0.5$, $\beta = 0.5$ and $\gamma = 0.1$, Eq. (7) aims to maximize $0.05\ (f_{gb} + f_{lb}) - 0.45\ (f_{gw} + f_{lw})$. This indicates that on the USPS data set, effective discriminative features can be obtained by maximizing $0.05(f_{gb} + f_{lb}) - 0.45(f_{gw} + f_{lw})$. This also indicates that when extracting discriminative features, the importance of f_{gb} and f_{lb} is the same, and the importance of f_{gw} and f_{lw} is the same. However, the importance of f_{gb} and f_{gw} is different, and the importance of f_{lb} and f_{lw} is different. This demonstrates that different types of dissimilarities have different importance in learning the projection matrix when the WAPL algorithm deals with different data sets. Ignoring the weights of any type of dissimilarities may undermine the ability of the projection matrix to extract discriminative features. Therefore, it is essential to control the weights of these dissimilarities based on the characteristics of data sets.

6.5 Results of the Approaches

6.5.1 Network Traffic

Fig. 9 presents that the proposed approach can reduce network traffic by nearly 93% compared with MEC_{simple} and MEC_{WAPL} approaches. The major reason is that, with the cloud-guided feature extraction approach, the edge server only needs to upload a small amount of effective discriminative feature data to the cloud server. However, in MEC_{simple} and MEC_{WAPL} approaches, the edge server uploads a large amount of feature data to the cloud server. This is because MEC_{simple} and MEC_{WAPL} approaches extract numerous features to preserve the intrinsic information of images.

In addition, Fig. 9 presents that the proposed approach can reduce network traffic by nearly 1000 times compared with MCC_{simple} and MCC_{WAPL} approaches. This is because

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2944371, IEEE Transactions on Mobile Computing

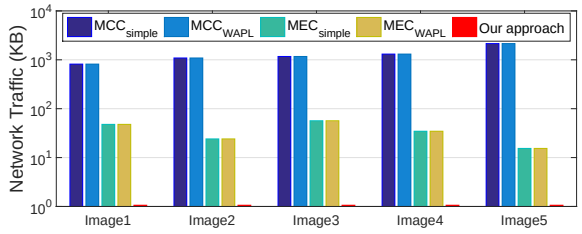


Fig. 9. Network traffic for different approaches.

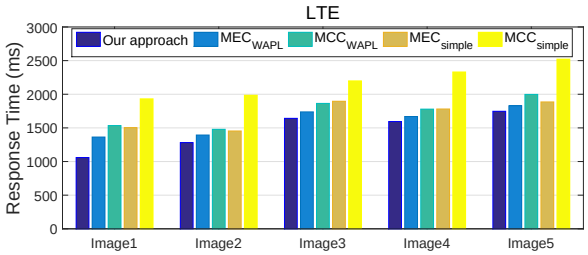


Fig. 10. Response time for different approaches.

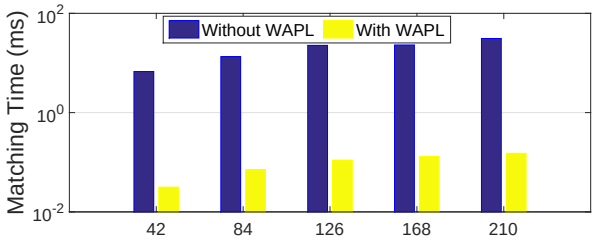


Fig. 11. Matching time for different cases.

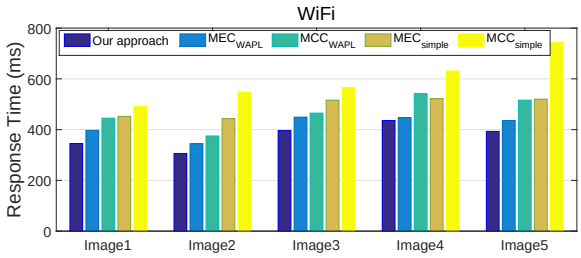


Fig. 12. Response time for different approaches.

the mobile device uploads the raw image data to the cloud server in MCC_{simple} and MCC_{WAPL} approaches. However, in the proposed approach, the edge server uploads discriminative feature data to the cloud server. Compared with the raw image data, the size of the discriminative feature data is much smaller. Therefore, the proposed approach can significantly reduce network traffic on the core network. Moreover, MCC_{simple} and MCC_{WAPL} approaches have the same network traffic because they upload the raw image data. MEC_{simple} and MEC_{WAPL} approaches have the same network traffic because they upload the feature data extracted by the LBP algorithm. Furthermore, we also observe that MEC_{simple} and MEC_{WAPL} approaches can reduce network traffic by 17 times compared with MCC_{simple} and MCC_{WAPL} approaches.

6.5.2 Response Time

Fig. 10 depicts the response time of five approaches when the mobile device connects to the edge server through LTE. We observe that the proposed approach can reduce the average response time by 35% compared with the MCC_{simple} approach. The reason is that the network transmission time is reduced because the edge server only uploads little feature data to the cloud server. In addition, feature matching time is reduced since the feature matching algorithm is performed in the low-dimensional feature data set space. Therefore, the average response time can be reduced.

Fig. 10 also depicts that the response time of MCC_{simple} approach is longer than the response time of MEC_{simple} approach. The reason is that MCC_{simple} approach uploads the raw image data to the cloud server, and MEC_{simple} approach uploads the extracted feature data by using the LBP algorithm to the cloud server. Also, the feature data is smaller than the raw image data. Therefore, the response time of MCC_{simple} approach is longer. The response time of the MCC_{WAPL} approach is longer than our approach because the network traffic of the MCC_{WAPL} approach is larger. Under the same bandwidth, the greater the network traffic is, the longer the network transmission delay is. Moreover, since MEC_{WAPL} approach performs the feature matching in the low-dimensional feature data set space, the response time of MEC_{simple} approach is longer than the response time of MEC_{WAPL} approach.

Fig. 11 reveals that using the WAPL algorithm significantly reduces the feature matching time since it can cut down the number of matching features. We also observe that the feature matching time can be reduced by 100 times compared with the case where the WAPL algorithm is not used. The major reason is that numerous features incur long matching time when the number of images is the same.

Our approach can get a minimum response time. The major reason is that our approach can extract a small number of effective discriminative features with the cloud-guided feature extraction. Hence, the network transmission time can be reduced since the discriminative features are small. In addition, the feature matching time can be reduced since the dimension of the feature data set space is low.

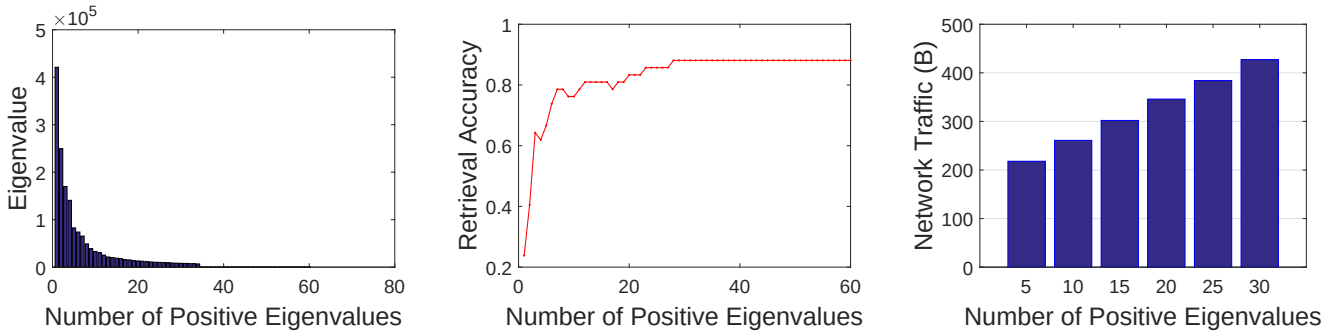
In addition, we also evaluate the response time of five approaches when the mobile device connects to the edge server through WiFi. Fig. 12 displays that the proposed approach can reduce the average response time by 47% compared with MCC_{simple} approach. Specifically, when our approach and MCC_{simple} approach recognize the Image5, the response time is 393 ms and 742 ms. This indicates that with the development of 5G technology, our approach can significantly reduce response time because the transmission rate between mobile devices and edge servers becomes faster.

6.5.3 Retrieval Accuracy

In this experiment, we randomly select 90% images from the Lab_face data set for training and the rest of images are used for testing.

Table 6 shows that the approaches using the WAPL algorithm achieve the accuracy of 88.33%, which is 6.9% higher than the approaches using the LBP algorithm. The reason is that the projection matrix learned by the WAPL algorithm can remove redundant features. In addition, the projection matrix can extract effective discriminative features from the image data set on the cloud server and the image on the edge server. However, the LBP algorithm aims to extract features to preserve the intrinsic structure of the image data,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2944371, IEEE Transactions on Mobile Computing



(a) (b) (c)

Fig. 13. Relationship between eigenvalue, retrieval accuracy, network traffic and the number of positive eigenvalues

TABLE 6
Comparison of Retrieval Accuracy on Lab_face Data Set

Approach	Retrieval Accuracy (%)
MCC_{simple}	81.43%
MCC_{WAPL}	88.33%
MEC_{simple}	81.43%
MEC_{WAPL}	88.33%
Our approach	**88.33%**

rather than extracting discriminative features for retrieval. Therefore, using the WAPL algorithm can achieve higher retrieval accuracy.

6.5.4 Interaction Strategy

Theorem 1 demonstrates that the projection matrix can extract effective discriminative features when it consists of eigenvectors corresponding to the positive eigenvalues. In this section, we investigate the relationship between eigenvalue, retrieval accuracy, network traffic and the number of positive eigenvalues. Fig. 13 exhibits that retrieval accuracy and network traffic increase as the number of positive eigenvalues increases. This indicates that the higher the accuracy is, the more network traffic is required.

For Scenario I, we can choose the number of positive eigenvalues as the dimension of the feature data set. The corresponding retrieval accuracy is 88.33% and the network traffic is 427 B. For Scenario II, we can choose 1/3 the number of positive eigenvalues as the dimension of the feature data set. The corresponding retrieval accuracy is 76.19% and the network traffic is 261 B. For scenario III, 1/2 the number of positive eigenvalues can be chosen as the dimension of the feature data set. The corresponding retrieval accuracy is 80.95% and network traffic is 302 B. The higher the retrieval accuracy is, the higher the dimension of the feature data set is required. However, higher dimensions of the feature data set result in larger network traffic and longer response time. Therefore, users can choose different interaction strategies in terms of retrieval accuracy and response time.

7 Discussion

The proposed approach uses the matrix $\mathbf{P}$ to extract discriminative features from image data sets on cloud servers and images on edge servers. Thus, the proposed approach can reduce the network traffic since the edge servers only upload the extracted discriminative features to cloud servers. Although the size of the matrix $\mathbf{P}$ is large, the proposed approach can save network traffic. The reason is that numerous users can use the matrix $\mathbf{P}$ to extract discriminative features. However, cloud servers only send the matrix $\mathbf{P}$ to edge servers once. For example, assume that the raw image data is 2 MB; the pre-processed image data is 100 KB; the feature data extracted by using the matrix $\mathbf{P}$ is 2 KB; the projection matrix is 5 MB. When there are 10,000 users requesting the image retrieval, edge servers upload the pre-processed image data to cloud servers with the network traffic of 1,000,000 KB. Edge servers upload the extracted discriminative feature data to cloud servers with the network traffic of 25,120 KB. The results show that uploading the extracted feature data can reduce network traffic. As the number of users requesting image retrieval increases, the proposed approach saves more network traffic. Moreover, network transmission delay can also be reduced since the network traffic is reduced. In addition, the dimension of the feature data set is equal to the dimension of $\mathbf{P}^T\mathbf{x}$. Performing the feature matching on the low-dimensional feature data set takes less time since the dimension of $\mathbf{P}^T\mathbf{x}$ is low. Therefore, the proposed approach consumes less network traffic and response time. In addition, the proposed approach can be applied to applications that generate image data or video data, such as virtual reality, augmented reality, and computer vision.

8 Conclusion

In this paper, we propose a cloud-guided feature extraction approach for image retrieval in the mobile edge computing environment. Our goal is to improve retrieval accuracy, reduce network traffic and response time, and meet the requirements of users. In the proposed approach, we propose a projection matrix learning algorithm to generate a projection matrix that guides the feature extraction from the image data set and the image to be retrieved. Hence, the network traffic and network transmission time can be reduced because the projection matrix can extract a small amount of effective discriminative features from images and edge servers only upload a small amount of feature data to cloud servers. In addition, the dimension of the feature data set is low. The response time can also be reduced since the feature matching algorithm is performed in the low-dimensional feature data set space. Therefore, the proposed approach can reduce the network traffic and response time. The advantages of the proposed approach have been demonstrated by a prototype system using a real MEC environment. However, the learned projection matrix in the current form is static. In order to make the projection

matrix extract effective discriminative features when a large number of new images appear, and automatically meet the user's requirements for accuracy and response time in different scenarios, we intend to extend this work in two directions in the future. The first is to model the retrieval accuracy, the number of eigenvalues and the response time to dynamically update the projection matrix. The second is to evaluate the scenarios and automatically select the appropriate accuracy and response time based on the requirements of these scenarios.

REFERENCES

[1] L. Zhu, J. L. Shen, L. Xie, and Z. Y. Cheng, "Unsupervised Visual Hashing with Semantic Assistant for Content-Based Image Retriveal," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 2, pp. 472-486, 2017.

[2] W. T. Xu, Y. R. Shen, N. Bergmann, and W. Hu, "Sensor-Assisted Multi-View Face Recognition System on Smart Glass," *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 197-210, 2018.

[3] M. Shahzad, A. X. Liu, and A. Samuel, "Behavior Based Human Authentication on Touch Screen Devices Using Gestures and Signatures," *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2726-2741, 2017.

[4] J. Zhang, Z. F. Zhang, and H. Guo, "Towards Secure Data Distribution Systems in Mobile Cloud Computing," *IEEE Transactions on Mobile Computing*, vol. 16, no. 11, pp. 3222-3235, 2017.

[5] Y. C. Liu, M. J. Lee, and Y. Y. Zheng, "Adaptive Multi-Resource Allocation for Cloudlet-Based Mobile Cloud Computing System," *IEEE Transactions on Mobile Computing*, vol. 15, no. 10, pp. 2398-2410, 2016.

[6] S. Yang, D. Kwon, H. Yi, Y. Cho, Y. Kwon, and Y. Paek, "Techniques to Minimize State Transfer Costs for Dynamic Execution Offloading in Mobile Cloud Computing," *IEEE Transactions on Mobile Computing*, vol. 13, no. 11, pp. 2648-2660, 2014.

[7] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog Computing May Help to Save Energy in Cloud Computing," *IEEE Journal on Selected Areas In Communications*, vol. 34, no. 5, pp. 1728-1739, 2016.

[8] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile Edge Computing: A Survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450-465, 2018.

[9] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. B. Heinzelman, "Cloud-Vision: Real-time Face Recognition Using a Mobile-Cloudlet-Cloud Acceleration Architecture," In *Proceedings of the IEEE Symposium on Computers and Communications*, pp. 59-66, 2012.

[10] P. F. Hu, H. S. Ning, T. Qiu, Y. F. Zhang, and X. Luo, "Fog Computing-Based Face Identification and Resolution Scheme in Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1910-1920, 2017.

[11] X. L. Liu, X. Xie, S. G. Wang, J. Liu, D. D. Yao, J. N. Cao, and K. Q. Li, "Efficient Range Queries for Large-scale Sensor-augmented RFID Systems," *IEEE/ACM Transactions on Networking*, in press, 2019.

[12] X. L. Liu, J. N. Cao, Y. N. Yang, W. Y. Qu, X. B. Zhao, K. Q. Li, and D. D. Yao,, "Fast RFID Sensory Data Collection: Trade-off Between Computation and Communication Costs," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1179-1191, 2019.

[13] Q. Fan, N. Ansari, "Towards Traffic Load Balancing in Drone-Assisted Communications for IoT," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3633-3640, 2019.

[14] S. Barman, H. P. H. Shum, S. Chattopadhyay, D. Samanta, "A Secure Authentication Protocol for Multi-Server-Based E-Healthcare Using a Fuzzy Commitment Scheme," *IEEE Access*. vol. 7, pp. 12557-12574, 2019.

[15] K. Jo, J. Kim, D. Kim, C. Jang, M. Sunwoo, "Development of Autonomous Car-Part II: A Case Study on the Implementation of an Autonomous Driving System Based on Distributed Architecture," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 8, pp. 5119-5132, 2015.

[16] T. Ojala, M. Pietikainen, and D. Harwood, "Performance Evaluation of Texture Measures with Classification Based on Kullback Discrimination of Distributions," In *Proceedings of the International Conference on Pattern Recognition*, pp. 582-585, 1994.

[17] W. G. Zhou, H. Q. Li, J. Sun, and Q. Tian, "Collaborative Index Embedding for Image Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1154-1166, 2018.

[18] X. S. Wei, J. H. Luo, J. X. Wu, and Z. H. Zhou, "Selective Convolutional Descriptor Aggregation for Fine-Grained Image Retrieval," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2868-2881, 2017.

[19] A. M. Martinez, and A. C. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228-233, 2001.

[20] R. Haeb-Umbach, and H. Ney, "Linear Discriminant Analysis for Improved Large Vocabulary Continuous Speech Recognition," In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 13-16, 1992.

[21] S. C. Yan, D. Xu, B. Y. Zhang, H. J. Zhang, Q. Yang, and S. Lin, "Graph Embedding and Extensions: A General Framework for Dimensionality Reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40-51, 2007.

[22] Q. X. Gao, J. J. Liu, H. L. Zhang, X. B. Gao, and K. Li, "Joint Global and Local Structure Discriminant Analysis," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 4, pp. 626-635, 2013.

[23] X. L. Li, M. L. Chen, F. P. Nie, and Q. Wang, "Locality Adaptive Discriminant Analysis," In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 2201-2207, 2017.

[24] B. L. Li, Q. Lu, and S. W. Yu, "An Adaptive K-nearest neighbor text categorization strategy," *ACM Transactions on Asian Language Information Processing*, vol. 3, no. 4, pp. 215-226, 2004.

[25] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin,"LIBLINEAR: A Library for Large Linear Classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871-1874, 2008.

[26] Y. Sarikaya, H. Inaltekin, T. Alpcan, and J. S. Evans, "Stability and Dynamic Control of Underlay Mobile Edge Networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 9, pp. 2195-2208, 2018.

[27] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao, and X. M. Shen, "Cooperative Edge Caching in User-Centric Clustered Mobile Networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1791-1805, 2018.

[28] C. Liu, Y. Cao, Y. Luo, G. L. Chen, V. Vokkarane, Y. S. Ma, S. Q. Chen, and P. Hou, "A New Deep Learning-based Food Recognition System for Dietary Assessment on An Edge Computing Service Infrastructure," *IEEE Transactions on Service Computing*, vol. 11, no. 2, pp. 249-261, 2018.

[29] U. Drolia, K. Guo, J. Q. Tan, R. Gandhi, and P. Narasimhan, "Cachier: Edge-caching for Recognition Applications," In *Proceedings of the International Conference on Distributed Computing Systems*, pp. 276-286, 2017.

[30] Y. X. Sun, S. Zhou, and J. Xu, "EMM:Energy-Aware Mobility Management for Mobile Edge Computing in Ultra Dense Networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2637-2646, 2017.

[31] X. Ma, S. Zhang, P. Yang, N. Zhang, C. Lin, and X. M. Shen, "Cost-Efficient Resource Provisioning in Cloud Assisted Mobile Edge Computing", In *Proceedings of the IEEE Global Communications Conference*, pp. 1-6, 2017.

[32] X. Chen, L. Jiao, W. Z. Li, and X. M. Fu, "Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795-2808, 2016.

[33] Y. Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic Computation Offloading for Mobile-Edge Computing with Energy Harvesting Devices," *IEEE Journal on Selected Areas In Communications*, vol. 34, no. 12, pp. 3590-3605, 2016.

[34] Y. Xiao, and M. Krunz, "QoE and Power Efficiency Tradeoff for Fog Computing Networks and Fog Node Cooperation," In *Proceedings of the IEEE International Conference on Computer Communications*, pp. 1-9, 2017.

[35] L. Tong, Y. Li, and W. Gao, "A Hierarchical Edge Cloud Architecture for Mobile Computing," In *Proceedings of the IEEE International Conference on Computer Communications*, pp. 1-9, 2016.

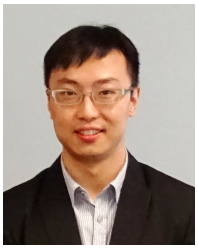
[36] A. Ceselli, M. Premoli, and S. Secci, "Mobile Edge Cloud Network Design Optimization," *IEEEACM Transactions on Networking*, vol. 25, no. 3, pp. 1818-1831, 2017.

[37] S. N. Shirazi, A. Gouglidis, A. Farshad, and D. Hutchison, "The Extended Cloud: Review and Analysis of Mobile Edge Computing and Fog From a Security and Resilience Perspective," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2586-2595, 2017.

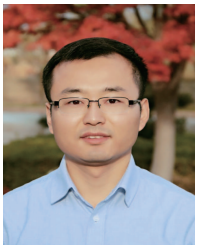
[38] P. A. Viola, and M. J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 511-518, 2001.
[39] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," In *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 849-856, 2001.
[40] C. T. Ding, and L. Zhang, "Double Adjacency Graphs-based Discriminant Neighborhood Embedding," *Pattern Recognition*, vol. 48, no. 5, pp. 1734-1742, 2015.
[41] G. H. Golub, and C. F. V. Loan, "Matrix Computations," *Johns Hopkins University Press Baltimore, USA (1996)*.
[42] OpenAirUsage, https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/OpenAirUsage, [Online; accessed Sepember 3, 2019].
[43] OpenAirInterface, https://www.openairinterface.org/, [Online; accessed Sepember 3, 2019].
[44] N. Nikaein, M. K. Marina, S. Manickam, A. Dowson, R. Knopp, and C. Bonnet, "OpenAirInterface: A Flexible Platform for 5G Research," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 33-38, 2014.
[45] F. Gringoli, P. Patras, C. Donato, P. Serrano, and Y. Grunenberger, "Performance Assessment of Open Software Platforms for 5G Prototyping," *IEEE Wireless Communications*, vol. 25, no. 5, pp. 10-15, 2018.
[46] A. Banerjee, R. Mahindra, K. Sundaresan, S. K. Kasera, K. V. D. Merwe, and S. Rangarajan, "Scaling the LTE Control-Plane for Future Mobile Access," In *Proceedings of the ACM Conference on Emerging Networking Experiments and Technologies*, pp. 1-13, 2015.
[47] S. Zimmo, A. Moubayed, A. Refaey, and A. Shami, "Coexistence of WiFi and LTE in the Unlicensed Band Using Time-Domain Virtualization," In *Proceedings of the IEEE Global Communications Conference*, pp. 1-6, 2018.
[48] X. J. Wang, T. Q. S. Quek, M. Sheng, and J. D. Li, "Throughput and Fairness Analysis of Wi-Fi and LTE-U in Unlicensed Band," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 1, pp. 63-78, 2017.
[49] A. Galanopoulos, F. Foukalas, and T. A. Tsiftsis, "Efficient Coexistence of LTE with WiFi in the Licensed and Unlicensed Spectrum Aggregation," *IEEE Transactions on Cognitive Communications and Networking*, vol. 2, no. 2, pp. 129-140, 2016.
[50] The Extended Yale Face Database B, http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html, [Online; accessed June 7, 2019].
[51] The UMIST Face Database, https://www.sheffield.ac.uk/eee/research/iel/research/face, [Online; accessed June 7, 2019].
[52] The USPS Handwritten Digits Database, https://riemenschneider.hayko.at/vision/dataset/task.php?did=96, [Online; accessed June 7, 2019].
[53] Lab_face, http://sguangwang.com/BUPT-facedataset.html, [Online; accessed June 7, 2019].

Shangguang Wang received his PhD degree at Beijing University of Posts and Telecommunications in 2011. He is a professor and Deputy Director at the State Key Laboratory of Networking and Switching Technology (BUPT). He has published more than 100 papers, and played a key role at many international conferences, such as general chair and PC chair. His research interests include service computing, cloud computing, and mobile edge computing. He is a senior member of the IEEE, and the Editor-in-Chief of the International Journal of Web Science.

Chuntao Ding received the B.S. and M.S. degrees from SIAS International University in 2012 and Soochow University in 2015, respectively, both in software engineering. He is currently a Ph.D. candidate at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. His research interests include machine learning, mobile edge computing.

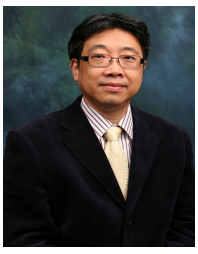


Ning Zhang is an Assistant Professor at Texas A&M University-Corpus Christi, USA. He received the Ph.D degree from University of Waterloo, Canada, in 2015. After that, he was a postdoc research fellow at University of Waterloo and University of Toronto, Canada, respectively. He serves/served as an associate editor of IEEE Access and IET Communication, an area editor of Encyclopedia of Wireless Networks (Springer) and Cambridge Scholars, a guest editor of Wireless Communication and Mobile Computing, International Journal of Distributed Sensor Networks, and Mobile Information System. He also served as the workshop chair for the first IEEE Workshop on Cooperative Edge. He is a recipient of the Best Paper Awards at IEEE Globecom 2014 and IEEE WCSP 2015, respectively. His current research interests include next generation mobile networks, physical layer security, machine learning, and mobile edge computing.



Xiulong Liu is currently a postdoctoral fellow in Department of Computing, Hong Kong Polytechnic University, Hong Kong, China. Before that, he received the B.E. degree and Ph.D. degree from the School of Software Technology and the School of Computer Science and Technology, Dalian University of Technology, China, in 2010 and 2016, respectively. His research interests include RFID systems and wireless sensor networks. He has published more than 30 research papers in prestigious journals and conferences including TON, TMC, TC, TPDS, INFOCOM, etc.

Ao Zhou received the Ph.D. degrees in Beijing University of Posts and Telecommunications, Beijing, China, in 2015. She is currently an Associate Professor with State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. She has published 20+ research papers. She played a key role at many international conferences. Her research interests include Cloud Computing and Edge Computing.



Jiannong Cao (M'93-SM'05-F'14) received the Ph.D. degree in computer science from Washington State University, Pullman, WA, USA, in 1990. He is currently a Chair Professor of Department of Computing at The Hong Kong Polytechnic University, Hong Kong. He is also the director of the Internet and Mobile Computing Lab in the department and the director of Universitys Research Facility in Big Data Analytics. His research interests include parallel and distributed computing, wireless sensing and networks, pervasive and mobile computing, and big data and cloud computing.

Xuemin (Sherman) Shen (M97, SM02, F09) received the B.Sc. (1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering. He is a Professor and University Research Chair, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is also the Associate Chair for Graduate Studies. Dr. Shen's research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He is an elected member of IEEE ComSoc Board of Governor, and the Chair of Distinguished Lecturers Selection Committee. Dr. Shen served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, the General Chair for ACM Mobihoc'15, the Symposia Chair for IEEE ICC'10, the Tutorial Chair for IEEE VTC'11 Spring and IEEE ICC'08, the Technical Program Committee Chair for IEEE Globecom'07, the General Co-Chair for Chinacom'07 and QShine'06, the Chair for IEEE Communications Society Technical Committee on Wireless Communications, and P2P Communications and Networking. He also serves/served as the Editor-in-Chief for IEEE Network, IEEE Internet of Things Journal, Peer-to-Peer Networking and Application, and IET Communications; a Founding Area Editor for IEEE Transactions on Wireless Communications; an Associate Editor for IEEE Transactions on Vehicular Technology, Computer Networks, and ACM/Wireless Networks, etc.; and the Guest Editor for IEEE JSAC, IEEE Wireless Communications, IEEE Communications Magazine, and ACM Mobile Networks and Applications, etc. Dr. Shen received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004, 2007 and 2010 from the University of Waterloo, the Premiers Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society.