Information Fusion 63 (2020) 1–12

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Deep feature fusion through adaptive discriminative metric learning for scene recognition

Chen Wang[a,b,*], Guohua Peng[a], Bernard De Baets[b]

[a] Department of Applied Mathematics Northwestern Polytechnical University, Xi'an, Shaanxi 710072, PR China
[b] KERMIT, Department of Data Analysis and Mathematical Modelling Ghent University, Coupure links 653, Ghent 9000, Belgium

ARTICLE INFO

Keywords:
Deep feature fusion
Adaptive discriminative metric learning
Scene recognition

ABSTRACT

With the development of deep learning techniques, fusion of deep features has demonstrated the powerful capability to improve recognition performance. However, most researchers directly fuse different deep feature vectors without considering the complementary and consistent information among them. In this paper, from the viewpoint of metric learning, we propose a novel deep feature fusion method, called deep feature fusion through adaptive discriminative metric learning (DFF-ADML), to explore the complementary and consistent information for scene recognition. Concretely, we formulate an adaptive discriminative metric learning problem, which not only fully exploits discriminative information from each deep feature vector, but also adaptively fuses complementary information from different deep feature vectors. Besides, we map different deep feature vectors of the same image into a common space by different linear transformations, such that the consistent information can be preserved as much as possible. Moreover, DFF-ADML is extended to a kernelized version. Extensive experiments on both natural scene and remote sensing scene datasets demonstrate the superiority and robustness of the proposed deep feature fusion method.

1. Introduction

Scene recognition, which aims to label an image according to a set of semantic categories, has attracted increasing attention in various computer vision tasks such as image retrieval [1], visual surveillance [2], and so on. Although various recognition approaches [3–5] have been proposed over the past few decades, it remains a challenging problem because of intra-class diversity and inter-class similarity in scene images. As can be seen in Fig. 1 (a) and (b), the second image in the class ‘Living room’ is easily misclassified in the class ‘Bedroom’ due to the high inter-class similarity of these classes. Traditional methods are mainly based on low-level features and mid-level features. The former directly extract the basic visual features of scene images, while the latter attempt to comprehensively describe a scene image by latent semantic information. Although these methods have produced good results for scene recognition, the lack of a more meaningful and abstractive scene representation greatly limits their recognition performance.

In recent years, deep convolutional neural networks (CNNs) have achieved a prominent performance in the domain of scene recognition due to the availability of large-scale image datasets and computer technology. Existing deep learning approaches can be divided into three directions: (1) pre-trained deep features [6]; (2) fine-tuned deep features [7]; (3) full-trained deep features [8,9]. Actually, for practical scene recognition tasks, it is hard to fully train a new deep CNN model from scratch. Therefore, most researchers focus on pre-trained deep features and try to exploit the deep features from convolutional layers and fully connected layers. For instance, Liu et al. [10] used deep convolutional features to learn a sparse representative and discriminative model consisting of multiple parts. Tang et al. [6] divided the GoogLeNet model into three parts of layers from bottom to top and applied the output features from each of the three parts for scene recognition. Xie et al. [11] constructed both a mid-level local representation and a convolutional Fisher vector representation based on dictionary learning, and integrated the CNN features from fully connected layers to obtain the complementary information. However, none of these methods pays attention to the fusion of different deep feature vectors. Inspired by the fact that different deep feature vectors possess unique representation powers, we firmly believe that it is very attractive to explore the complementary and consistent information among them.

Metric learning has become one of the most popular tools to solve various machine learning problems [12]. The essence of metric learning is to find a transformation that allows to transform the original sample into a more representative and discriminative feature space. Depending on how the sample information is exploited, metric learning can be cat-

[*] Corresponding author.
E-mail addresses: chenchenwang@mail.nwpu.edu.cn (C. Wang), penggh@nwpu.edu.cn (G. Peng), bernard.debaets@ugent.be (B. De Baets).

https://doi.org/10.1016/j.inffus.2020.05.005
Received 27 December 2019; Received in revised form 8 May 2020; Accepted 10 May 2020
Available online 14 May 2020

Fig. 1. Two classes of the Scene-15 dataset: (a) Living room; (b) Bedroom.

egorized into global metric learning and local metric learning. In the global view, Metric Learning with Side Information [13], Information-theoretic Metric Learning [14], and Mahalanobis Metric Learning for Clustering [15] have been widely used in many computer vision tasks. In the local view, representative methods are Neighborhood Component Analysis [16] and Large Margin Nearest Neighbor [17]. Alternatively, other researchers integrated global and local metric learning into a unified learning framework [18–20], which is helpful to some extent to obtain a more reasonable distance metric.

In this paper, taking advantage of metric learning, we propose a novel deep feature fusion method for scene recognition. More specifically, we first extract multiple deep feature vectors from pre-trained CNN models. Then, an adaptive discriminative metric learning problem is formulated, which can simultaneously exploit discriminative information from each deep feature vector and adaptively fuse complementary information from different deep feature vectors. Besides, we map different deep feature vectors of the same image into a common space by different linear transformations, such that the consistent information can be preserved as much as possible. As a result, the proposed deep feature fusion method has the potential to learn the complementary and consistent information among different deep feature vectors, thereby improving the performance of scene recognition. The main contributions of our work can be summarized as follows:

(1) We propose a deep feature fusion method through adaptive discriminative metric learning. To the best of our knowledge, it is the first time that metric learning has been introduced into deep feature fusion for handling the problem of scene recognition.
(2) An alternating iterative strategy is devised to solve the corresponding optimization problem effectively. Moreover, the proposed method is extended to a kernelized version for more complex problems.
(3) Extensive experiments on both natural scene and remote sensing scene datesets demonstrate the superiority and robustness of the proposed deep feature fusion method.

The remainder of this paper is organized as follows. Related work is presented in Section 2. Section 3 introduces the proposed deep feature fusion through adaptive discriminative metric learning. Experimental results are given in Section 4. Section 5 concludes this paper.

2. Related work

In this section, we briefly review two related topics: deep feature fusion and scene recognition.

2.1. Deep feature fusion

The deep features from convolutional layers exhibit meaningful local structural information, while those from fully connected (FC) layers represent rich global semantic information. Accordingly, most researchers have devoted attention to the fusion of convolutional features or FC-features. Khan et al. [21] proposed to transform the structured convolutional activations to another highly discriminative feature space, so as to exploit rich mid-level convolutional features. Yang et al. [22] presented a part-based CNN model to optimize and select discriminative mid-level visual elements, which were applied to multiple layers of a pre-trained CNN to obtain more diverse visual elements. Guo et al. [23] studied an efficient Fisher convolutional vector (FCV) that successfully rescues the orderless mid-level semantic information. Then, both the FCV-and FC-features were collaboratively employed in a novel locally supervised deep hybrid model. Ye et al. [24] put forward a parallel multi-stage architecture formed by a low, middle and high deep convolutional neural network sub-model to automatically learn representative and discriminative hierarchical features. Several others tried to fuse different deep feature vectors. Yu and Liu [25] adopted two feature fusion strategies to fuse two deep convolutional feature vectors extracted from the original RGB stream and the saliency stream. Sun et al. [26] fused deep features extracted from three discriminative views including the information of object semantics, global appearance and contextual appearance. However, none of these methods explored the complementary and consistent information among different deep feature vectors, which limits the recognition performance to some extent.

2.2. Scene recognition

Existing scene recognition methods can be divided into three categories based on the features used: low-level features, mid-level features and high-level features. Low-level features mainly describe color, texture, or structure information to characterize the local visual representation. Examples are Local Binary Patterns (LBP) [27] and the Scale Invariant Feature Transform (SIFT) [28]. To alleviate the semantic gap between low-level features and high-level abstract semantics, mid-level features were developed. Bag of Visual Words (BOVW) [29] is one of the most successful models for scene recognition. Along this line, Spatial Pyramid Matching (SPM) [30] was further developed by integrating the spatial information. Fisher Vectors [31] make use of the Gaussian mixture model to produce more statistical information. Owing to the development of CNN, high-level deep features are capable of generating more abstractive and meaningful scene representations, thus resulting in state-of-the-art recognition performance. Among various CNN models, CaffeNet [32], AlexNet [33], GoogLeNet [34], VGGNet [35], and ResNet [36] are widely known because of their enhanced representation power and superior performance. More recently, mainstream recognition methods [11,23] focus on combining the deep features from convolutional layers and fully connected layers, most of which have been detailed in Section 2.1. Besides, some other studies [6,37] use representative CNN models to improve the recognition performance. Liu et al. [37] combined ResNet-based transfer learning and data augmentation. In our work, we will aggregate three representative deep feature vectors to explore deep feature fusion from the viewpoint of metric learning.

3. Deep feature fusion through adaptive discriminative metric learning

In this section, we first introduce the proposed DFF-ADML in detail, and then extend it to a kernelized version to deal with more complex problems. Finally, we conduct the corresponding complexity and convergence analysis.

3.1. Deep feature extraction

For practical scene recognition tasks, the limited availability of training images makes it difficult to fully train a new deep CNN model from scratch. Besides, several studies have demonstrated that the deep features of images rarely depend on the final application. Therefore, a pretrained CNN model can be employed as a feature extractor for any image. In our work, we employ three representative CNN models to extract three deep feature vectors.

GoogLeNet. In 2014, GoogLeNet won the first prize in the ImageNet competition. It uses inception modules to obtain a deeper network and avoid over-fitting. We extract the features of the fully connected layer as scene representation, which results in a vector of 1024 dimensions.

VGGNet. VGGNet, who won the second prize in the same competition as GoogLeNet, also became prominent in many real-world applications. We select VGGNet-16 as feature extractor, which contains 13 convolutional layers, 5 pooling layers, and 3 fully connected layers. We extract the features of the first fully connected layer, which results in a vector of 4096 dimensions.

ResNet. In 2015, ResNet won the first prize in the ImageNet competition. It is characterized by the design of a block in the form of a 'bottleneck'. Specifically, the model of ResNet-152 contains 50 building blocks with each block consisting of 3 layers, and 1 fully connected layer at the end. We extract the features of the fully connected layer and thus obtain a vector of 2048 dimensions.

3.2. Problem formulation

Let $\mathcal{F} = \{\left((\mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^V), l_i\right) \mid i = 1, \ldots, n\}$ represent the set of deep features extracted from the training images, where $\mathbf{x}_i^\upsilon \in \mathbb{R}^{d_\upsilon}(\upsilon = 1, \ldots, V)$ represents the v-th feature vector of the i-th training image and $l_i \in \{1, 2, \ldots, C\}$ (C is the number of classes of scene images) stands for the associated label; d_v denotes the dimensionality of the v-th feature vector and n is the total number of training images. To overcome the intra-class diversity and inter-class similarity of scene images, we try to learn a discriminative distance metric such that the distance between samples of the same class is as small as possible, while the distance between samples of different classes is as large as possible. Most studies pay particular attention to the Mahalanobis distance metric because it is conveniently optimized [38]. For the v-th feature vector, let $\mathbf{M}_\upsilon \in \mathbb{R}^{d_\upsilon \times d_\upsilon}$ denote a symmetric positive semi-definite matrix, which is used to parametrize the Mahalanobis distance metric. In order to fully explore the discriminative information, we use each training sample with the associated label and formulate the discriminative metric learning problem as

$$\min_{\mathbf{M}_\upsilon} \sum_{i=1}^{n} \sum_{j : l_i = l_j} d^2_{\mathbf{M}_\upsilon}(\mathbf{x}_i^\upsilon, \mathbf{x}_j^\upsilon) - \sum_{i=1}^{n} \sum_{j : l_i \neq l_j} d^2_{\mathbf{M}_\upsilon}(\mathbf{x}_i^\upsilon, \mathbf{x}_j^\upsilon), \tag{1}$$

where the distance between $\mathbf{x}_i^\upsilon$ and $\mathbf{x}_j^\upsilon$ is computed as

$$d^2_{\mathbf{M}_\upsilon}(\mathbf{x}_i^\upsilon, \mathbf{x}_j^\upsilon) = (\mathbf{x}_i^\upsilon - \mathbf{x}_j^\upsilon)^{\mathrm{T}} \mathbf{M}_\upsilon (\mathbf{x}_i^\upsilon - \mathbf{x}_j^\upsilon). \tag{2}$$

Since the matrix $\mathbf{M}_\upsilon$ is positive semi-definite, it can be decomposed as $\mathbf{M}_\upsilon = \mathbf{W}_\upsilon \mathbf{W}_\upsilon^{\mathrm{T}}$ (for the dimensions of $\mathbf{W}_\upsilon$, see further on). Eq. (2) can be rewritten as

$$\begin{aligned} d^2_{\mathbf{W}_\upsilon}(\mathbf{x}_i^\upsilon, \mathbf{x}_j^\upsilon) &= (\mathbf{x}_i^\upsilon - \mathbf{x}_j^\upsilon)^{\mathrm{T}} \mathbf{W}_\upsilon \mathbf{W}_\upsilon^{\mathrm{T}} (\mathbf{x}_i^\upsilon - \mathbf{x}_j^\upsilon) \\ &= \|\mathbf{W}_\upsilon^{\mathrm{T}} \mathbf{x}_i^\upsilon - \mathbf{W}_\upsilon^{\mathrm{T}} \mathbf{x}_j^\upsilon\|^2. \end{aligned} \tag{3}$$

This implies that metric learning can be viewed as learning a linear transformation, which transforms the deep feature vectors into a more discriminative feature space.

A large number of works have demonstrated that multiple deep feature vectors can provide richer information than a single deep feature vector. Different deep feature vectors characterize the scene image from different points of view, thus these deep feature vectors are able to provide complementary information. However, how to explore and fuse the complementary information from different deep feature vectors remains a challenging problem. The adaptive fusion strategy [39] fuses different feature vectors by learning the corresponding adaptive weights, thus having the ability to exploit the complementary information of different feature vectors. Taking advantage of the flexibility and generalization ability of adaptive fusion, we fuse different deep feature vectors through the following adaptive discriminative metric learning problem

$$\begin{aligned} \min_{\mathbf{W}, \boldsymbol{\alpha}} & \sum_{\upsilon=1}^{V} \alpha_\upsilon \left(\sum_{i=1}^{n} \sum_{j : l_i = l_j} \|\mathbf{W}_\upsilon^{\mathrm{T}} \mathbf{x}_i^\upsilon - \mathbf{W}_\upsilon^{\mathrm{T}} \mathbf{x}_j^\upsilon\|^2 - \sum_{i=1}^{n} \sum_{j : l_i \neq l_j} \|\mathbf{W}_\upsilon^{\mathrm{T}} \mathbf{x}_i^\upsilon - \mathbf{W}_\upsilon^{\mathrm{T}} \mathbf{x}_j^\upsilon\|^2 \right) \\ & + \beta \sum_{\upsilon=1}^{V} \|\mathbf{W}_\upsilon\|_F^2 \\ & \text{s.t.} \ \sum_{\upsilon=1}^{V} \alpha_\upsilon = 1, \ \alpha_\upsilon \geq 0, \end{aligned} \tag{4}$$

where $\mathbf{W} = [\mathbf{W}_1^{\mathrm{T}}, \mathbf{W}_2^{\mathrm{T}}, \ldots, \mathbf{W}_V^{\mathrm{T}}]^{\mathrm{T}}$ is the transformation matrix, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_V]\alpha_V]$ is the adaptive weight vector, and β is the regularization parameter to prevent the entries in the matrices $\mathbf{W}_\upsilon$ from being too large. If the solution to problem (4) is $\boldsymbol{\alpha} = [0, \ldots, 0, 1, 0, \ldots, 0]$, only one deep feature vector is kept, which deviates from the idea of feature fusion. Motivated by [40], we modify α_v to be α_υ^r, where $r > 1$ guarantees that more than one deep feature vector is selected so that the complementary information can be well employed. Then, the objective function is rewritten as

$$\begin{aligned} \min_{\mathbf{W}, \boldsymbol{\alpha}} & \sum_{\upsilon=1}^{V} \alpha_\upsilon^r \left(\sum_{i=1}^{n} \sum_{j : l_i = l_j} \|\mathbf{W}_\upsilon^{\mathrm{T}} \mathbf{x}_i^\upsilon - \mathbf{W}_\upsilon^{\mathrm{T}} \mathbf{x}_j^\upsilon\|^2 - \sum_{i=1}^{n} \sum_{j : l_i \neq l_j} \|\mathbf{W}_\upsilon^{\mathrm{T}} \mathbf{x}_i^\upsilon - \mathbf{W}_\upsilon^{\mathrm{T}} \mathbf{x}_j^\upsilon\|^2 \right) \\ & + \beta \sum_{\upsilon=1}^{V} \|\mathbf{W}_\upsilon\|_F^2 \\ & \text{s.t.} \ \sum_{\upsilon=1}^{V} \alpha_\upsilon = 1, \ \alpha_\upsilon \geq 0. \end{aligned} \tag{5}$$

It is worth noting that adaptive discriminative metric learning not only fully exploits discriminative information from each deep feature vector, but also adaptively fuses complementary information from different deep feature vectors.

While each transformation matrix $\mathbf{W}_\upsilon$ in problem (5) exhibits enhanced discriminative power, different such matrices may not be consistent with each other. Actually, different deep feature vectors characterize the same scene image, and hence should be closely correlated in the learned metric spaces. To this end, we attempt to map the different deep feature vectors into a common space. Considering that different deep feature vectors usually have a different dimensionality, we use different transformation matrices $\mathbf{W}_\upsilon$, where $\mathbf{W}_\upsilon \in \mathbb{R}^{d_\upsilon \times m}$ represents the v-th transformation matrix and m denotes the mapped dimensionality in the common space. After that, we can minimize the discrepancy between two different deep feature vectors of the same image as

$$\min_{\mathbf{W}} \sum_{\upsilon, l=1}^{V} d^2_{\mathbf{W}_\upsilon, \mathbf{W}_l}(\mathbf{x}_i^\upsilon, \mathbf{x}_i^l), \tag{6}$$

where

$$\begin{aligned} d^2_{\mathbf{W}_\upsilon, \mathbf{W}_l}(\mathbf{x}_i^\upsilon, \mathbf{x}_i^l) &= (\mathbf{W}_\upsilon^{\mathrm{T}} \mathbf{x}_i^\upsilon - \mathbf{W}_l^{\mathrm{T}} \mathbf{x}_i^l)^{\mathrm{T}} (\mathbf{W}_\upsilon^{\mathrm{T}} \mathbf{x}_i^\upsilon - \mathbf{W}_l^{\mathrm{T}} \mathbf{x}_i^l) \\ &= \|\mathbf{W}_\upsilon^{\mathrm{T}} \mathbf{x}_i^\upsilon - \mathbf{W}_l^{\mathrm{T}} \mathbf{x}_i^l\|^2. \end{aligned} \tag{7}$$

For the entire set of training images, we have

$$\min_{\mathbf{W}} \sum_{i=1}^{n} \sum_{v,l=1}^{V} \|\mathbf{W}_v^{\mathrm{T}}\mathbf{x}_i^v - \mathbf{W}_l^{\mathrm{T}}\mathbf{x}_i^l\|^2. \tag{8}$$

In this way, the different deep feature vectors are consistent with each other in the common space, thereby sufficiently exploring the consistent information among the different deep feature vectors.

Finally, the objective function of DFF-ADML is formulated as

$$\begin{aligned}
&\min_{\mathbf{W},\boldsymbol{\alpha}} \sum_{v=1}^{V} \alpha_v^r \left(\sum_{i=1}^{n} \sum_{j:l_i=l_j} \|\mathbf{W}_v^{\mathrm{T}}\mathbf{x}_i^v - \mathbf{W}_v^{\mathrm{T}}\mathbf{x}_j^v\|^2 - \sum_{i=1}^{n} \sum_{j:l_i\neq l_j} \|\mathbf{W}_v^{\mathrm{T}}\mathbf{x}_i^v - \mathbf{W}_v^{\mathrm{T}}\mathbf{x}_j^v\|^2 \right) \\
&\quad + \beta \sum_{v=1}^{V} \|\mathbf{W}_v\|_F^2 + \eta \sum_{i=1}^{n} \sum_{v,l=1}^{V} \|\mathbf{W}_v^{\mathrm{T}}\mathbf{x}_i^v - \mathbf{W}_l^{\mathrm{T}}\mathbf{x}_i^l\|^2 \\
&\quad \text{s.t. } \mathbf{W}_v^{\mathrm{T}}\mathbf{W}_v = \mathbf{I}, v = 1, \ldots, V, \ \sum_{v=1}^{V} \alpha_v = 1, \alpha_v \geq 0,
\end{aligned} \tag{9}$$

where $\mathbf{W}_v^{\mathrm{T}}\mathbf{W}_v = \mathbf{I} \in \mathbb{R}^{m\times m}$ is set to avoid degenerate solutions [15]. $\mathbf{I}$ is an m-dimensional identity matrix and η is a regularization parameter.

After solving optimization problem (9), we can obtain the corresponding transformation matrix $\mathbf{W}_v$ for each deep feature vector, which has the ability to transform the pre-trained deep feature vectors into a more discriminative feature space where the complementary and consistent information is fully explored. Combining these discriminative feature vectors with corresponding weight coefficients α_ν, we can generate the ultimate fused features for the i-th image as

$$\mathbf{x}_i' = \sum_{v=1}^{V} \alpha_v \mathbf{W}_v^{\mathrm{T}}\mathbf{x}_i^v. \tag{10}$$

3.3. Optimization procedure

Given the non-linear optimization problem in (9), solving for the variables $\mathbf{W}$ and $\boldsymbol{\alpha}$ simultaneously is intractable by directly applying gradient descent. $\mathbf{W}$ changes along with $\boldsymbol{\alpha}$, and vice versa. We solve this problem with an effective alternating iterative strategy, so that the optimal transformation matrix $\mathbf{W}$ and the adaptive weight vector $\boldsymbol{\alpha}$ can be jointly learned. Before that, we derived a simplified expression

$$\begin{aligned}
&\sum_{i=1}^{n} \sum_{j:l_i=l_j} \|\mathbf{W}_v^{\mathrm{T}}\mathbf{x}_i^v - \mathbf{W}_v^{\mathrm{T}}\mathbf{x}_j^v\|^2 - \sum_{i=1}^{n} \sum_{j:l_i\neq l_j} \|\mathbf{W}_v^{\mathrm{T}}\mathbf{x}_i^v - \mathbf{W}_v^{\mathrm{T}}\mathbf{x}_j^v\|^2 \\
&\quad = \sum_{i=1}^{n} \sum_{j=1}^{n} \|\mathbf{W}_v^{\mathrm{T}}\mathbf{x}_i^v - \mathbf{W}_v^{\mathrm{T}}\mathbf{x}_j^v\|^2 \mathbf{S}_{ij}^w - \sum_{i=1}^{n} \sum_{j=1}^{n} \|\mathbf{W}_v^{\mathrm{T}}\mathbf{x}_i^v - \mathbf{W}_v^{\mathrm{T}}\mathbf{x}_j^v\|^2 \mathbf{S}_{ij}^b \\
&\quad = 2\mathrm{tr}\big(\mathbf{W}_v^{\mathrm{T}}\mathbf{X}^v\mathbf{L}^w(\mathbf{X}^v)^{\mathrm{T}}\mathbf{W}_v\big) - 2\mathrm{tr}\big(\mathbf{W}_v^{\mathrm{T}}\mathbf{X}^v\mathbf{L}^b(\mathbf{X}^v)^{\mathrm{T}}\mathbf{W}_v\big) \\
&\quad = 2\mathrm{tr}\big(\mathbf{W}_v^{\mathrm{T}}\mathbf{X}^v(\mathbf{L}^w - \mathbf{L}^b)(\mathbf{X}^v)^{\mathrm{T}}\mathbf{W}_v\big) \\
&\quad = 2\mathrm{tr}\big(\mathbf{W}_v^{\mathrm{T}}\mathbf{R}_v\mathbf{W}_v\big),
\end{aligned} \tag{11}$$

where $\mathbf{R}_v = \mathbf{X}^v(\mathbf{L}^w - \mathbf{L}^b)(\mathbf{X}^v)^{\mathrm{T}}$ and $\mathbf{X}^v = [\mathbf{x}_1^v, \mathbf{x}_2^v, \ldots, \mathbf{x}_n^v]$. $\mathbf{S}_{ij}^w$ is defined as $\mathbf{S}_{ij}^w = 1$, if $l_i = l_j$, and $\mathbf{S}_{ij}^w = 0$ otherwise. $\mathbf{S}_{ij}^b$ is defined as $\mathbf{S}_{ij}^b = 1$, if $l_i \neq l_j$, and $\mathbf{S}_{ij}^b = 0$ otherwise. Furthermore, $\mathbf{L}^w = \mathbf{D}^w - \mathbf{S}^w$ denotes the Laplacian matrix of the label matrix $\mathbf{S}^w$, and $\mathbf{D}^w$ is a diagonal matrix given by $\mathbf{D}_{ii}^w = \sum_{j=1}^n \mathbf{S}_{ij}^w$. $\mathbf{L}^b = \mathbf{D}^b - \mathbf{S}^b$ denotes the Laplacian matrix of the label matrix $\mathbf{S}^b$, and $\mathbf{D}^b$ is a diagonal matrix given by $\mathbf{D}_{ii}^b = \sum_{j=1}^n \mathbf{S}_{ij}^b$.

Updating $\mathbf{W}$. Given the initial $\boldsymbol{\alpha}$, we compute $\mathbf{W}$. The objective function in (9) can be rewritten as

$$\begin{aligned}
&\min_{\mathbf{W}} \sum_{v=1}^{V} \alpha_v^r \mathrm{tr}\big(\mathbf{W}_v^{\mathrm{T}}\mathbf{R}_v\mathbf{W}_v\big) + \beta \sum_{v=1}^{V} \mathrm{tr}\big(\mathbf{W}_v^{\mathrm{T}}\mathbf{W}_v\big) \\
&\quad + \eta \sum_{i=1}^{n} \mathrm{tr}\left([\mathbf{W}_1^{\mathrm{T}}, \mathbf{W}_2^{\mathrm{T}}, \ldots, \mathbf{W}_V^{\mathrm{T}}] \mathbf{X}_i\mathbf{L}\mathbf{X}_i^{\mathrm{T}} \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \vdots \\ \mathbf{W}_V \end{bmatrix} \right) \\
&\quad \text{s.t. } \mathbf{W}_v^{\mathrm{T}}\mathbf{W}_v = \mathbf{I}, v = 1, \ldots, V.
\end{aligned} \tag{12}$$

Here $\mathbf{X}_i = \mathrm{diag}(\mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^V)$ and $\mathbf{L}$ is a Laplacian matrix defined as $\mathbf{L} = \mathbf{D} - \mathbf{1}_{V\times V}$, where $\mathbf{D}$ is a diagonal matrix given by $\mathbf{D}_{ii} = \sum_{j=1}^V (\mathbf{1}_{V\times V})_{ij}$ and $\mathbf{1}_{V\times V}$ is a $V \times V$ matrix with all ones.

Since $\mathbf{W} = [\mathbf{W}_1^{\mathrm{T}}, \mathbf{W}_2^{\mathrm{T}}, \ldots, \mathbf{W}_V^{\mathrm{T}}]^{\mathrm{T}}$, the problem (12) can be further rewritten as

$$\begin{aligned}
&\min_{\mathbf{W}} \mathrm{tr}\big(\mathbf{W}^{\mathrm{T}}(\mathbf{R} + \beta\mathbf{I} + \eta\mathbf{Q})\mathbf{W}\big) \\
&\quad \text{s.t. } \frac{1}{V}\mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I},
\end{aligned} \tag{13}$$

where $\mathbf{R} = \mathrm{diag}(\alpha_1^r\mathbf{R}_1, \alpha_2^r\mathbf{R}_2, \ldots, \alpha_V^r\mathbf{R}_V)$ and $\mathbf{Q} = \sum_{i=1}^n \mathbf{X}_i\mathbf{L}\mathbf{X}_i^{\mathrm{T}}$. Consequently, the solution $\mathbf{W}$ to problem (9) can be obtained by solving the following eigen-decomposition problem

$$V(\mathbf{R} + \beta\mathbf{I} + \eta\mathbf{Q})\mathbf{W} = \boldsymbol{\Lambda}\mathbf{W}, \tag{14}$$

where $\boldsymbol{\Lambda}$ is a Lagrangian multiplier. Thus $\mathbf{W}$ consists of the eigenvectors corresponding to the first m smallest eigenvalues of the matrix $V(\mathbf{R} + \beta\mathbf{I} + \eta\mathbf{Q})$.

Updating $\boldsymbol{\alpha}$. With fixed $\mathbf{W}$, we update $\boldsymbol{\alpha}$. The objective function in (9) leads to the following optimization problem

$$\begin{aligned}
&\min_{\boldsymbol{\alpha}} \sum_{v=1}^{V} \alpha_v^r \mathrm{tr}\big(\mathbf{W}_v^{\mathrm{T}}\mathbf{R}_v\mathbf{W}_v\big) \\
&\quad \text{s.t. } \sum_{v=1}^{V} \alpha_v = 1, \ \alpha_v \geq 0.
\end{aligned} \tag{15}$$

Following the Lagrange multiplier method, the Lagrange function is constructed as

$$L(\boldsymbol{\alpha}, \lambda) = \sum_{v=1}^{V} \alpha_v^r \mathrm{tr}\big(\mathbf{W}_v^{\mathrm{T}}\mathbf{R}_v\mathbf{W}_v\big) - \lambda\left(\sum_{v=1}^{V} \alpha_v - 1\right), \tag{16}$$

where λ is a Lagrange multiplier. Setting $\frac{\partial L(\boldsymbol{\alpha},\lambda)}{\partial \alpha_v} = 0$ and $\frac{\partial L(\boldsymbol{\alpha},\lambda)}{\partial \lambda} = 0$, we get

$$\begin{cases} r\alpha_v^{r-1}\mathrm{tr}\big(\mathbf{W}_v^{\mathrm{T}}\mathbf{R}_v\mathbf{W}_v\big) - \lambda = 0 \\ \sum_{v=1}^V \alpha_v - 1 = 0. \end{cases} \tag{17}$$

Thus, we can obtain α_ν as

$$\alpha_v = \frac{\big(1/\mathrm{tr}\big(\mathbf{W}_v^{\mathrm{T}}\mathbf{R}_v\mathbf{W}_v\big)\big)^{1/(r-1)}}{\sum_{v=1}^V \big(1/\mathrm{tr}\big(\mathbf{W}_v^{\mathrm{T}}\mathbf{R}_v\mathbf{W}_v\big)\big)^{1/(r-1)}}. \tag{18}$$

We iterate the above procedure until the algorithm converges. A simplified pseudo-code implementation of DFF-ADML is summarized in Algorithm 1.

3.4. Kernelized version

Linear metric learning can work well under the linearity assumption, whereas it is not powerful enough for more complex problems [41]. To overcome this limitation, we extend DFF-ADML to a kernelized version by using the kernel trick, and propose deep feature fusion through adaptive kernel discriminative metric learning (DFF-AKDML).

We first map the ν-th feature vector into a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ via a feature map ϕ_ν with corresponding kernel function $\mathrm{K}^v(\mathbf{x}_i^v, \mathbf{x}_j^v) = \langle \phi_v(\mathbf{x}_i^v), \phi_v(\mathbf{x}_j^v) \rangle_{\mathcal{H}}$ [42]. After that, the feature vector in $\mathcal{H}$ is mapped into $\mathbb{R}^m$ by a linear transformation $\mathbf{P}_v\colon \mathcal{H} \to \mathbb{R}^m$. Because the linear transformation $\mathbf{P}_v$ should lie in the span of $\phi_v(\mathbf{x}_1^v), \phi_v(\mathbf{x}_2^v), \ldots, \phi_v(\mathbf{x}_n^v)$, there exists a transformation matrix $\mathbf{A}_v$ such that $\mathbf{P}_v = \boldsymbol{\Phi}_v\mathbf{A}_v$, where $\boldsymbol{\Phi}_v = [\phi_v(\mathbf{x}_1^v), \phi_v(\mathbf{x}_2^v), \ldots, \phi_v(\mathbf{x}_n^v)]$. Let $\mathbf{K}^v = \boldsymbol{\Phi}_v^{\mathrm{T}}\boldsymbol{\Phi}_v$, then the distance between the ν-th feature vector of two different images can be reformulated as

$$\begin{aligned}
d_{\mathbf{A}_v}^2(\phi_v(\mathbf{x}_i^v), \phi_v(\mathbf{x}_j^v)) &= \|\mathbf{P}_v^{\mathrm{T}}\phi_v(\mathbf{x}_i^v) - \mathbf{P}_v^{\mathrm{T}}\phi_v(\mathbf{x}_j^v)\|^2 \\
&= \|\mathbf{A}_v^{\mathrm{T}}\boldsymbol{\Phi}_v^{\mathrm{T}}\phi_v(\mathbf{x}_i^v) - \mathbf{A}_v^{\mathrm{T}}\boldsymbol{\Phi}_v^{\mathrm{T}}\phi_v(\mathbf{x}_j^v)\|^2 \\
&= \|\mathbf{A}_v^{\mathrm{T}}\mathbf{K}_{\cdot i}^v - \mathbf{A}_v^{\mathrm{T}}\mathbf{K}_{\cdot j}^v\|^2,
\end{aligned} \tag{19}$$

Algorithm 1 DFF-ADML.

```
Input:
    The set of deep features $\mathcal{F} = \{((\mathbf{x}_i^1, \mathbf{x}_i^2, \dots, \mathbf{x}_i^V), l_i) \mid i = 1, \dots, n\}$ for training images;
    The mapped dimensionality $m$;
    The tuning parameters $\beta$, $\eta$ and $r$.
Output:
    The transformation matrix $\mathbf{W}$, adaptive weight vector $\boldsymbol{\alpha}$ and fused feature matrix $\mathbf{X}' = [\mathbf{x}_1', \mathbf{x}_2', \dots, \mathbf{x}_n']$.
Initialize $\boldsymbol{\alpha} = [1/V, 1/V, \dots, 1/V]$;
Calculate $\mathbf{R}_v$, $v = 1, 2, \dots, V$;
Calculate $\mathbf{Q} = \sum_{i=1}^{n} \mathbf{X}_i \mathbf{L} \mathbf{X}_i^{\mathrm{T}}$;
while no convergence do
    Calculate $\mathbf{R} = \mathrm{diag}(\alpha_1^r \mathbf{R}_1, \alpha_2^r \mathbf{R}_2, \dots, \alpha_V^r \mathbf{R}_V)$;
    Solve the eigen-decomposition problem in Eq. (14) and obtain $\mathbf{W} = [\mathbf{W}_1^{\mathrm{T}}, \mathbf{W}_2^{\mathrm{T}}, \dots, \mathbf{W}_V^{\mathrm{T}}]^{\mathrm{T}}$;
    Update $\boldsymbol{\alpha}$ using Eq. (18);
end while
Get the fused features $\mathbf{x}_i' = \sum_{v=1}^{V} \alpha_v \mathbf{W}_v^{\mathrm{T}} \mathbf{x}_i^v$, $i = 1, 2, \dots, n$;
return $\mathbf{W}$, $\boldsymbol{\alpha}$, $\mathbf{x}_i'$;
```

where $\mathbf{K}_{.i}^v$ is the i-th column of the kernel matrix $\mathbf{K}^v$. Similarly, the distance between two different feature vectors of the same image can be reformulated as

$$d^2_{\mathbf{A}_v, \mathbf{A}_l}(\phi_v(\mathbf{x}_i^v), \phi_l(\mathbf{x}_i^l)) = \|\mathbf{A}_v^{\mathrm{T}} \mathbf{K}_{.i}^v - \mathbf{A}_l^{\mathrm{T}} \mathbf{K}_{.i}^l\|^2. \tag{20}$$

Based on the linear DFF-ADML, we can reformulate problem (9) in a kernelized version

$$\begin{aligned} \min_{\mathbf{A}, \boldsymbol{\alpha}^*} \sum_{v=1}^{V} \alpha_v^{*r} \left(\sum_{i=1}^{n} \sum_{j: l_i = l_j} \|\mathbf{A}_v^{\mathrm{T}} \mathbf{K}_{.i}^v - \mathbf{A}_v^{\mathrm{T}} \mathbf{K}_{.j}^v\|^2 - \sum_{i=1}^{n} \sum_{j: l_i \neq l_j} \|\mathbf{A}_v^{\mathrm{T}} \mathbf{K}_{.i}^v - \mathbf{A}_v^{\mathrm{T}} \mathbf{K}_{.j}^v\|^2 \right) \\ + \beta^* \sum_{v=1}^{V} \|\mathbf{A}_v\|_F^2 + \eta^* \sum_{i=1}^{n} \sum_{v,l=1}^{V} \|\mathbf{A}_v^{\mathrm{T}} \mathbf{K}_{.i}^v - \mathbf{A}_l^{\mathrm{T}} \mathbf{K}_{.i}^l\|^2 \\ \text{s.t. } \mathbf{A}_v^{\mathrm{T}} \mathbf{A}_v = \mathbf{I}, v = 1, \dots, V, \ \sum_{v=1}^{V} \alpha_v^* = 1, \alpha_v^* \geq 0, \end{aligned} \tag{21}$$

where $\mathbf{A} = [\mathbf{A}_1^{\mathrm{T}}, \mathbf{A}_2^{\mathrm{T}}, \dots, \mathbf{A}_V^{\mathrm{T}}]^{\mathrm{T}}$ is the transformation matrix, $\boldsymbol{\alpha}^* = [\alpha_1^*, \alpha_2^*, \dots, \alpha_V^*]$ is the adaptive weight vector, and β^* and η^* are two regularization parameters. $\mathbf{A}_v^{\mathrm{T}} \mathbf{A}_v = \mathbf{I} \in \mathbb{R}^{m \times m}$ is set to avoid degenerate solutions.

Intuitively, it can be seen that the form of DFF-AKDML is consistent with that of linear DFF-ADML. Therefore, the optimization procedure of the transformation matrix $\mathbf{A}$ and adaptive weight vector $\boldsymbol{\alpha}^*$ is similar to that for linear DFF-ADML. To be more specific, the solution $\mathbf{A}$ to optimization problem (21) consists of the eigenvectors corresponding to the first m smallest eigenvalues of the matrix $V(\mathbf{R}^* + \beta^* \mathbf{I} + \eta^* \mathbf{Q}^*)$, where $\mathbf{R}^* = \mathrm{diag}(\alpha_1^{*r} \mathbf{R}_1^*, \alpha_2^{*r} \mathbf{R}_2^*, \dots, \alpha_V^{*r} \mathbf{R}_V^*)$, $\mathbf{R}_v^* = \mathbf{K}^v(\mathbf{L}^w - \mathbf{L}^b)(\mathbf{K}^v)^{\mathrm{T}}$, $\mathbf{Q}^* = \sum_{i=1}^{n} \mathbf{K}_i \mathbf{L} \mathbf{K}_i^{\mathrm{T}}$ and $\mathbf{K}_i = \mathrm{diag}(\mathbf{K}_{.i}^1, \mathbf{K}_{.i}^2, \dots, \mathbf{K}_{.i}^V)$. Similarly, the weight coefficients are calculated as $\alpha_v^* = \frac{(1/\mathrm{tr}(\mathbf{A}_v^{\mathrm{T}} \mathbf{R}_v^* \mathbf{A}_v))^{1/(r-1)}}{\sum_{v=1}^{V} (1/\mathrm{tr}(\mathbf{A}_v^{\mathrm{T}} \mathbf{R}_v^* \mathbf{A}_v))^{1/(r-1)}}$.

3.5. Complexity and convergence analysis

Since DFF-ADML and DFF-AKDML apply a similar optimization procedure, we only analyze the computational complexity of DFF-ADML. According to Algorithm 1, the computational complexity can be calculated from three steps. First, the computational complexity of $\mathbf{R}$ and $\mathbf{Q}$ is $O(dn^2 + d^2 n)$ and $O(Vd^2 n)$, respectively, where $d = \Sigma_{v=1}^{V} d_v$. Second, the computational complexity for eigen-decomposition problem (14) is $O(d^3)$. Third, the computational cost for updating $\boldsymbol{\alpha}$ is about $O(md^2)$. Thus, the entire computational complexity of DFF-ADML is $O(\max(n, Vd)dn + Td^3)$, where T is the number of training iterations. It is worth noting that usually a low value of T (say at most 5) suffices, which will be demonstrated in the experiments section.

We show that the optimization procedure in Algorithm 1 monotonically reduces the objective function value. For simplicity, we denote the objective function (9) as $\mathcal{O}(\mathbf{W}, \boldsymbol{\alpha})$. According to the updating rules, with $\boldsymbol{\alpha}^t$ fixed, we have $\mathcal{O}(\mathbf{W}^{t+1}, \boldsymbol{\alpha}^t) \leq \mathcal{O}(\mathbf{W}^t, \boldsymbol{\alpha}^t)$. If $\mathbf{W}^{t+1}$ is fixed, we get $\mathcal{O}(\mathbf{W}^{t+1}, \boldsymbol{\alpha}^{t+1}) \leq \mathcal{O}(\mathbf{W}^{t+1}, \boldsymbol{\alpha}^t)$. Thus, we have $\mathcal{O}(\mathbf{W}^{t+1}, \boldsymbol{\alpha}^{t+1}) \leq \mathcal{O}(\mathbf{W}^t, \boldsymbol{\alpha}^t)$. It is easy to conclude that the objective function monotonically decreases and the corresponding iterative algorithm will converge to a local optimum.

4. Experiments

To evaluate the effectiveness of DFF-ADML and DFF-AKDML for scene recognition, we conduct experiments on both natural scene and remote sensing scene datasets. First, we introduce the datasets and experimental setup. Second, we conduct a parameter analysis to evaluate the impact of each parameter in the proposed method. Third, we compare our methods with different fusion methods as well as the state-of-the-art scene recognition methods. Finally, we conduct a convergence study to verify the efficiency of the proposed algorithm.

4.1. Datasets and experimental setup

Scene-15 [30]. This dataset includes 4485 images from 15 outdoor and indoor scene classes, with the number of images in each class ranging from 200 to 400. Sample images of each class are shown in Fig. 2. The average resolution of these images is 300×250 pixels. Based on the standard setting, we use 100 images per class as training images, and the remaining images per class as testing images.

MIT-67 [43]. This dataset contains 15,620 images of 67 indoor classes. The number of images varies, with at least 100 images per class. Sample images are shown in Fig. 3. Each image has a minimum resolution of 200 pixels on the smallest axis. We followed the standard evaluation protocol, randomly selecting 80 images from each class for training and 20 images for testing.

UCM-21 [44]. This dataset contains 2100 images from 21 high-resolution remote sensing classes, each class containing 100 images. Sample images are shown in Fig. 4. The resolution of these high-resolution images is 256 × 256 pixels. For fair comparison, we use 80 images per class as training images, and the remaining 20 images per class as testing images.

According to previous experience, for the Scene-15 and MIT-67 datasets, we use three deep models pre-trained on the scene-centric Place dataset [9] to extract three deep feature vectors, while for the UCM-21 dataset, we use those deep models pre-trained on the object-centric ImageNet dataset [45] to extract three deep feature vectors. To improve the computational efficiency of our proposed methods, the dimensionality of each feature vector is first reduced by PCA to preserve 99% energy. The resulting dimensionalities are denoted by d_i', $i = 1, \dots, V$. Once these reduced feature vectors are obtained, we can train DFF-ADML to generate the corresponding fused features, which are directly fed into an extreme learning machine (ELM), a classifier with a good recognition performance and a low computational cost. From here on, when we talk about the performance of DFF-ADML and DFF-AKDML, we tacitly refer to the performance of ELM trained on the basis of the fused features generated by these methods. The experiments are repeated 10 times with the training data and testing data of each class randomly selected, and the average recognition accuracies are taken as the final recognition accuracies. As for the kernel function in DFF-AKDML, the Gaussian kernel is adopted. To achieve the optimal recognition performance, the kernel parameter σ is selected from {0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 12.8, 25.6, 51.2}.

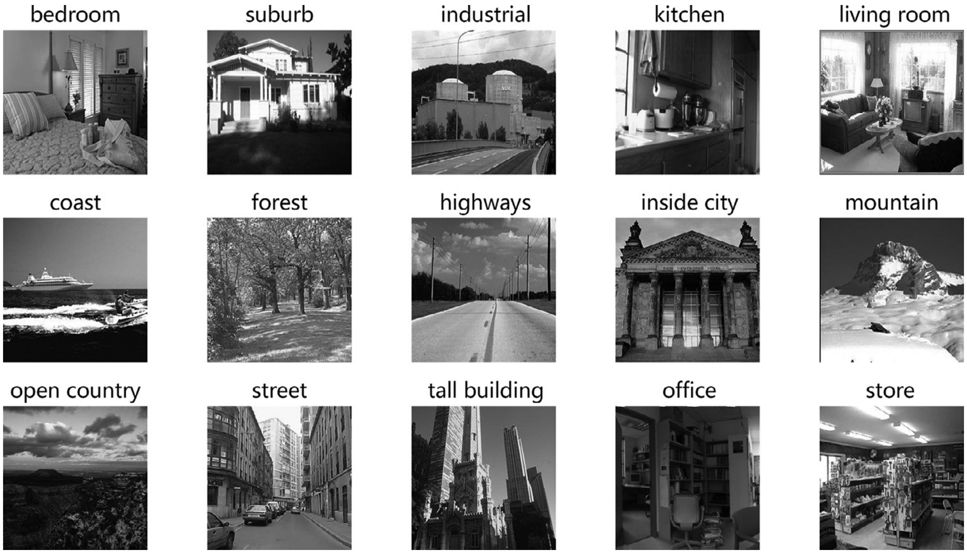


Fig. 2. Sample images of the Scene-15 dataset.

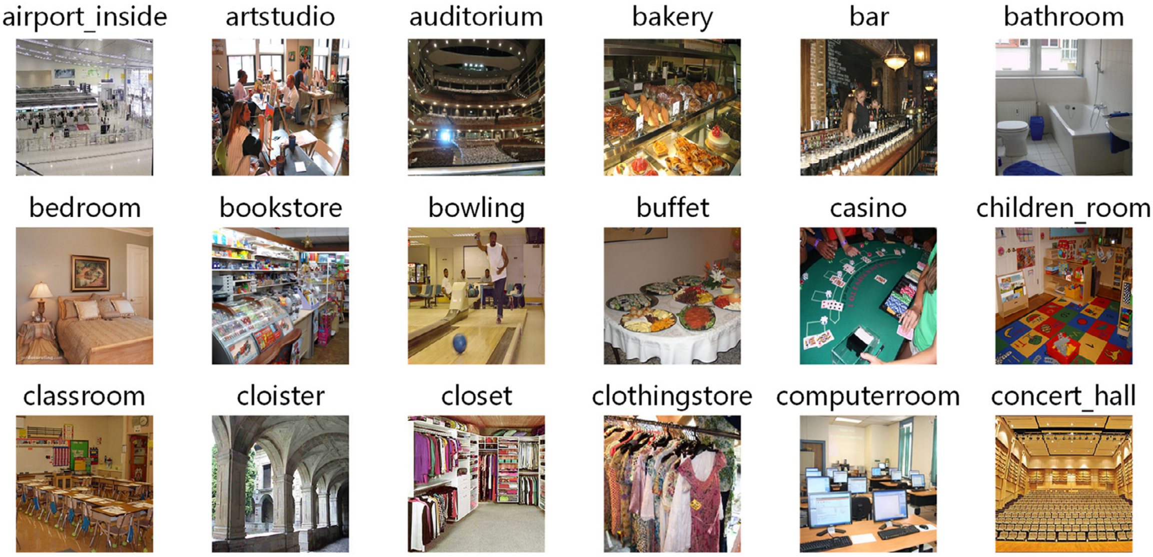


Fig. 3. Sample images of the MIT-67 dataset.

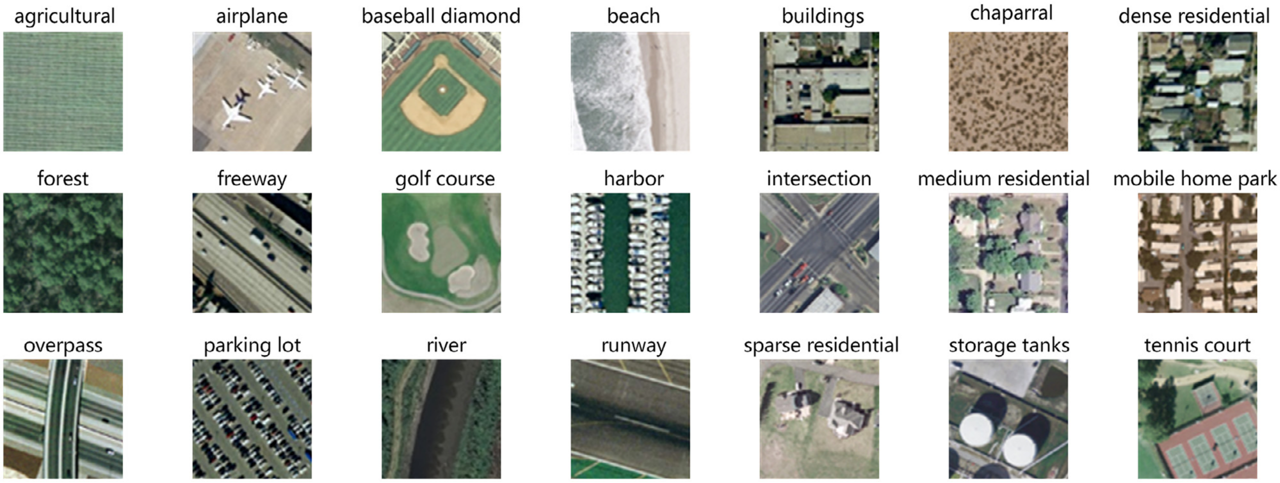


Fig. 4. Sample images of the UCM-21 dataset.

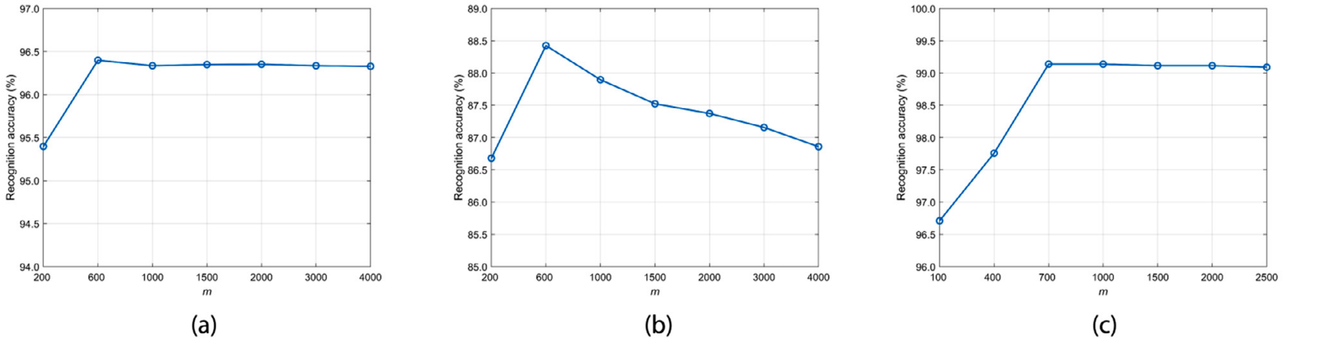


Fig. 5. Recognition accuracy with respect to the mapped dimensionality m on the three scene datasets. (a) Scene-15; (b) MIT-67; (c) UCM-21.

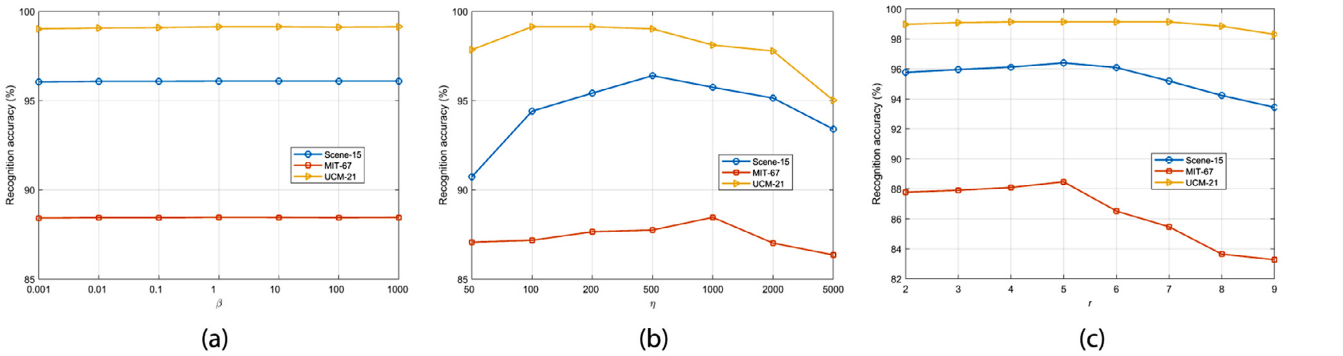


Fig. 6. Recognition accuracy with different tuning parameters on the three scene datasets. (a) parameter β; (b) parameter η; (c) parameter r.

4.2. Parameter analysis

For the sake of simplicity, we only analyze the parameters of DFF-ADML on the three scene datasets, since DFF-AKDML can be analyzed similarly. In order to apply DFF-ADML, essentially four parameters need to be set: the mapped dimensionality m, the regularization parameter β, the parameter η allowing to balance the complementary and consistent information and the parameter r allowing to make sure more than one deep feature vector is selected. Since tuning four parameters is quite challenging, we have adopted a pragmatic approach. Extensive explorative experiments have indicated that the recognition accuracy is not very sensitive to the two parameters β and r; this will be illustrated further on. We therefore decided to first fix these two parameters at $\beta = 1$ and $r = 5$, and then tune the other two parameters m and η alternately until convergence. We noticed that the optimal value of m was always situated around the smallest of the dimensionalities of the PCA-reduced feature vectors involved, and we thus systematically set $m = \min(d'_1, d'_2, \dots, d'_V)$. For the parameter η, we obtained as optimal values $\eta = 500$, $\eta = 1000$ and $\eta = 100$ for the Scene-15, MIT-67 and UCM-21 datasets, respectively.

For the sake of analyzing the impact of each parameter in DFF-ADML, we fix three of the four parameters as the above optimal values and report the recognition accuracy by varying a single parameter only [46]. First, we evaluate the effect of different values of the mapped dimensionality m. As shown in Fig. 5 (a) and (c), for the Scene-15 and UCM-21 datasets, the recognition accuracies of DFF-ADML rise gradually with an increase in the number of dimensions and up to a relative saturation point. In Fig. 5 (b), for the MIT-67 dataset, DFF-ADML achieves an optimal recognition accuracy when the mapped dimensionality is around 600. The results support our choice $m = \min(d'_1, d'_2, \dots, d'_V)$. One plausible explanation is that the consistent information among the different deep feature vectors can be well preserved with lower dimensionality.

Next, we evaluate the effect of the other three tuning parameters β, η and r. More specifically, the value of β is varied in the set {0.001, 0.01, 0.1, 1, 10, 100, 1000} and the recognition results are displayed in Fig. 6 (a). As announced earlier, the recognition accuracy of DFF-ADML is not sensitive to this parameter, indicating that DFF-ADML can obtain a robust recognition performance for a wide range of values of β. Furthermore, we vary η in the set {50, 100, 200, 500, 1000, 2000, 5000} and report the experimental results in Fig. 6 (b). As can be seen, the performance of DFF-ADML initially increases and then starts to decrease when η becomes too large, i.e. the value of η should not be too large or too small because it controls the consistent information among the different deep feature vectors. Hence, we can conclude that exploring consistent information is beneficial to improve the performance of scene recognition. Finally, the value of r is varied in the set {2, 3, 4, 5, 6, 7, 8, 9} and the recognition accuracies are presented in Fig. 6 (c). The recognition accuracy on the MIT-67 dataset is more sensitive than that on the other two datasets, as the result of more complex spatial layouts present in the MIT-67 dataset [7]. This confirms the selection of the consensus value $r = 5$.

4.3. Comparison and analysis of results

Comparison with different fusion strategies. We first investigate DFF-ADML and DFF-AKDML for two deep feature vectors and compare their recognition performance. From Tables 1 and 2, we draw the following conclusions. (1) DFF-ADML (GoogleNet + ResNet) and DFF-AKDML (GoogleNet + ResNet) consistently perform bet-

Table 1
Recognition accuracies (%) of DFF-ADML for two deep feature vectors.

Method	Scene-15	MIT-67	UCM-21
DFF-ADML (GoogleNet+VGGNet)	94.54	84.57	97.14
DFF-ADML (GoogleNet+ResNet)	95.42	85.62	97.86
DFF-ADML (VGGNet+ResNet)	95.91	86.78	98.31
DFF-ADML	**96.39**	**88.43**	**99.14**

Table 2
Recognition accuracies (%) of DFF-AKDML for two deep feature vectors.

Method	Scene-15	MIT-67	UCM-21
DFF-AKDML (GoogleNet+VGGNet)	94.59	83.34	97.06
DFF-AKDML (GoogleNet+ResNet)	95.04	85.23	97.45
DFF-AKDML (VGGNet+ResNet)	95.47	86.19	98.02
DFF-AKDML	**96.12**	**87.07**	**98.93**

Table 3
Recognition accuracies (%) of DFF-AKDML with different kernel functions.

Method	Scene-15	MIT-67	UCM-21
DFF-AKDML (Linear)	95.28	86.45	98.33
DFF-AKDML (Polynomial)	96.07	86.93	**99.12**
DFF-AKDML	**96.12**	**87.07**	98.93

Table 4
Recognition accuracies (%) of DFF-ADML with two classical fusion methods.

Method	Scene-15	MIT-67	UCM-21
CDFF-DML	95.52	86.96	98.07
PDFF-DML	94.21	85.11	97.61
DFF-ADML	**96.39**	**88.43**	**99.14**

ter than DFF-ADML (GoogleNet + VGGNet) and DFF-AKDML (GoogleNet + VGGNet), respectively. This is because ResNet extracts a more meaningful representation than VGGNet [36]. (2) DFF-ADML (VGGNet + ResNet) and DFF-AKDML (VGGNet + ResNet) outperform DFF-ADML (GoogleNet + ResNet) and DFF-AKDML (GoogleNet + ResNet), respectively, as the result of more deep scene information generated from VGGNet [35]. (3) Both DFF-ADML and DFF-AKDML achieve the best recognition performance by fusing three deep feature vectors. The results confirm that the proposed methods can effectively exploit those different deep feature vectors, thus having the ability to improve the recognition performance to some extent. (4) The performance of DFF-AKDML is inferior to that of DFF-ADML, since we only apply a Gaussian kernel to generate the fused features. For fair comparison, we compare the performance of DFF-AKDML versus different kernel functions on the three scene datasets. The experimental results are summarized in Table 3. DFF-AKDML with the linear kernel performs slightly worse than that with the polynomial kernel and the Gaussian kernel, while DFF-AKDML with the polynomial kernel achieves competitive recognition result than that with the Gaussian kernel, especially for the UCM-21 dataset. However, the polynomial kernel requires more parameters, which results in a higher computational cost.

Next, we compare the DFF-ADML method with two classical fusion strategies for three deep feature vectors:

- Concatenated deep feature fusion through discriminative metric learning (CDFF-DML): we concatenate three deep feature vectors into a long feature vector, and then use discriminative metric learning (see Eq. (4), where only one feature vector is kept) to obtain corresponding discriminative feature vectors as the fused features.
- Parallel deep feature fusion through discriminative metric learning (PDFF-DML): we use discriminative metric learning (see Eq. (4), where only one feature vector is kept) to obtain corresponding discriminative feature vectors for each deep feature vector, and then combine these feature vectors with equal weights to generate the fused features.

Table 4 reports the comparison results for three different fusion strategies. The following conclusions can be drawn. (1) For the three datasets, DFF-ADML performs better than CDFF-DML, which means that learning discriminative information from each deep feature vector is beneficial to improve the recognition performance. (2) The performance of DFF-ADML is superior to that of PDFF-DML. In particular, DFF-ADML gains a 3.3% improvement over PDFF-DML for the MIT-67 dataset, indicating that the parallel fusion method fails to explore the complementary and consistent information for scene recognition. (3) On the whole, DFF-ADML achieves the best recognition performance. The results demonstrate that our method not only helps to exploit discriminative information from each deep feature vector, but also adaptively fuses complementary information from different deep feature vectors.

Table 5
Recognition accuracies (%) of DFF-ADML with three widely used classifiers.

Method	Scene-15	MIT-67	UCM-21
DFF-ADML (KNN)	95.25	85.34	98.03
DFF-ADML (RF)	95.44	86.79	97.38
DFF-ADML (SVM)	96.38	87.68	98.57
DFF-ADML	**96.39**	**88.43**	**99.14**

Table 6
Performance comparison with the state-of-the-art methods on the Scene-15 dataset.

Method	Scene-15 (%)
ScSPM [3]	80.28
Object Bank [47]	80.90
ISPM [48]	83.30
ImageNet-AlexNet [9]	84.05
DDSFL [49]	84.42
ImageNet-GoogLeNet [9]	84.95
ImageNet-VGGNet [9]	86.28
Places365-AlexNet [9]	89.25
URDL [10]	91.15
Places365-GoogLeNet [9]	91.25
Places365-VGGNet [9]	91.97
DSFL+CNN [50]	92.81
G-MS2F [6]	92.90
FTOTLM [37]	94.01
Khan et al. [21]	94.50
SDO+fc features [5]	95.88
DFF-ADML	**96.39**

Comparison with different classifiers. To further evaluate the effectiveness of the proposed deep feature fusion method, we compare the performance of DFF-ADML with that of different classifiers for scene recognition. Apart from the ELM classifier, three widely used classifiers are employed, i.e., k-nearest neighbors (KNN), random forest (RF) and support vector machine (SVM). Table 5 reports the comparison results for the four classifiers, from which we can draw the following conclusions. (1) The performance of the SVM and ELM classifiers is superior to that of the KNN and RF classifiers. The main reason lies in the fact that SVM and ELM have trained more elaborate classification models based on the learned low-dimensional discriminative features, thus achieving a better recognition performance. (2) SVM achieves an almost comparable recognition performance as ELM, while ELM has a lower computational cost. (3) On the three scene datasets, we can see that DFF-ADML is able to suit different classifiers well, which demonstrates the robustness and effectiveness of the proposed deep feature fusion method. In particular, ELM always outperforms KNN, RF and SVM in terms of recognition accuracy.

Comparison with scene recognition methods. We compare DFF-ADML with a number of state-of-the-art scene recognition methods. The experimental results are summarized in Tables 6, 7 and 8 for the Scene-15, MIT-67, and UCM-21 datasets, respectively. From these tables, we can draw the following conclusions. (1) Methods based on high-level deep features are always superior to methods based on low-level and mid-level features. For example, AlexNet gains a higher recognition accuracy than OTC+HOG and DDSFL for the MIT-67 dataset. The results prove that deep CNN models have the ability to generate more semanti-

Table 7
Performance comparison with the state-of-the-art methods on the MIT-67 dataset.

Method	MIT-67 (%)
OTC+HOG [51]	47.33
DDSFL [49]	52.26
ImageNet-AlexNet [9]	56.79
ImageNet-GoogLeNet [9]	59.48
IFV+BOP [52]	63.10
ImageNet-VGGNet [9]	64.87
Places205-AlexNet [9]	68.24
Hybrid-CNN [8]	70.80
URDL [10]	71.90
FTOTLM [37]	74.63
DSFL+CNN [50]	76.23
Places205-GoogLeNet [9]	75.14
G-MS2F [6]	79.63
Places205-VGGNet [9]	79.76
Xie et al. [11]	82.24
Guo et al. [23]	83.75
Wang et al. [7]	86.70
SDO+fc features [5]	86.76
DFF-ADML	**88.43**

cally meaningful scene information. (2) The performance of DFF-ADML is superior to that of the other baseline deep CNN models, so it is effective to learn more informative deep features by discriminative metric learning, which therefore improves the performance of scene recognition. (3) DFF-ADML outperforms other fusion methods. Concretely, for the Scene-15 dataset, DFF-ADML yields a 3.5% higher accuracy than G-MS2F. For the MIT-67 dataset, DFF-ADML gains a 8.8% improvement over G-MS2F and an almost 1.7% improvement over SDO + fc features. For the UCM-21 dataset, DFF-ADML yields an almost 3.7% higher accuracy than LGF. The results demonstrate that our method actually makes a great contribution to adaptively fuse complementary information, while it has the potential to preserve the consistent information. (4) On both natural scene and remote sensing scene datasets, DFF-ADML achieves superior recognition accuracies, which also confirms that the proposed deep feature fusion method is more effective and robust.

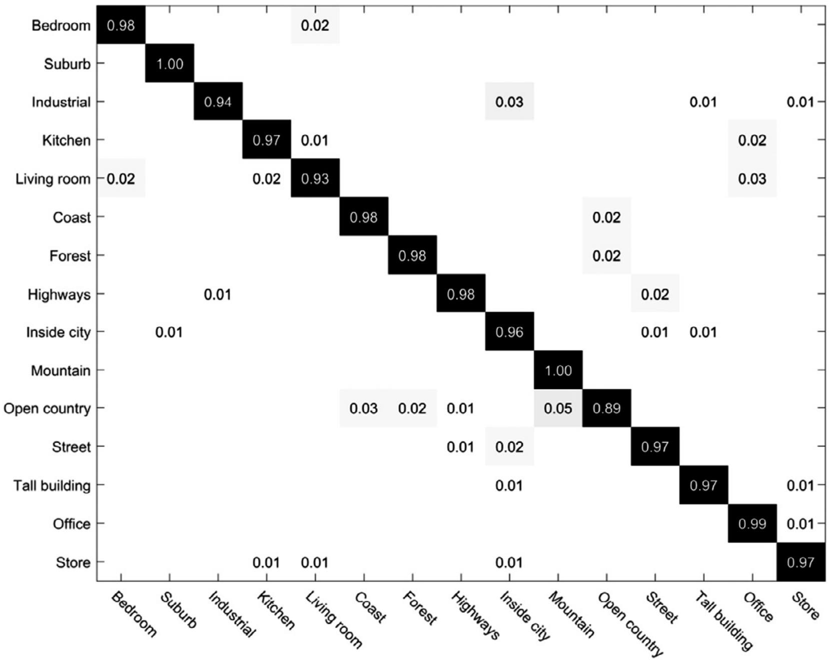
Additionally, in order to detail the recognition accuracies for each class, Figs. 7, 8 and 9 show the confusion matrices of DFF-ADML on the Scene-15, MIT-67 and UCM-21 datasets, respectively. As shown in Fig. 7, some of the images in the class ‘open country’ are more easily classified into the class ‘mountain’. The main reason is that part of the image information in the class ‘open country’ is similar to that of the class ‘mountain’. In Fig. 8, it can be seen that the diagonal values for the classes ‘deil’ and ‘museum’ are relatively low, but the diagonal values for several other classes are extremely high, such as ‘cloister’, ‘inside bus’, and ‘meeting room’. In Fig. 9, given the diagonal elements, most of the scene categories achieve satisfactory recognition results, except for some of the images in the class ‘buildings’ misclassified into the class ‘dense residential’ or ‘storage tanks’. To sum up, the recognition results demonstrate that DFF-ADML holds great potential to learn complementary and consistent information among the different deep feature vectors, and hence improve the performance of scene recognition.

Table 8
Performance comparison with the state-of-the-art methods on the UCM-21 dataset.

Method	UCM-21 (%)
LBP [53]	36.29
GIST [53]	46.90
BOVW(LBP) [53]	77.12
IFK(CH) [53]	83.79
GoogleNet [53]	94.31
VGGNet [53]	95.21
LGF [4]	95.48
ResNet+GMM [54]	96.67
Yu and Liu [25]	98.02
PMS [24]	98.81
DFF-ADML	**99.14**

4.4. Convergence study

To verify the efficiency of DFF-ADML, we also investigate the recognition accuracy in terms of the number of iterations on the Scene-15, MIT-67, and UCM-21 datasets. Fig. 10 (a)-(c) show the recognition performance of DFF-ADML over 20 iterations. We can see that the performance of DFF-ADML converges very quickly, usually within 5 iterations.



	Bedroom	Suburb	Industrial	Kitchen	Living room	Coast	Forest	Highways	Inside city	Mountain	Open country	Street	Tall building	Office	Store
Bedroom	0.98				0.02										
Suburb		1.00													
Industrial			0.94						0.03				0.01		0.01
Kitchen				0.97	0.01									0.02	
Living room	0.02			0.02	0.93									0.03	
Coast						0.98					0.02				
Forest							0.98				0.02				
Highways			0.01					0.98				0.02			
Inside city		0.01							0.96			0.01	0.01		
Mountain										1.00					
Open country						0.03	0.02	0.01		0.05	0.89				
Street								0.01	0.02			0.97			
Tall building									0.01				0.97		0.01
Office														0.99	0.01
Store				0.01	0.01				0.01						0.97

Fig. 7. Confusion matrix of DFF-ADML on the Scene-15 dataset.

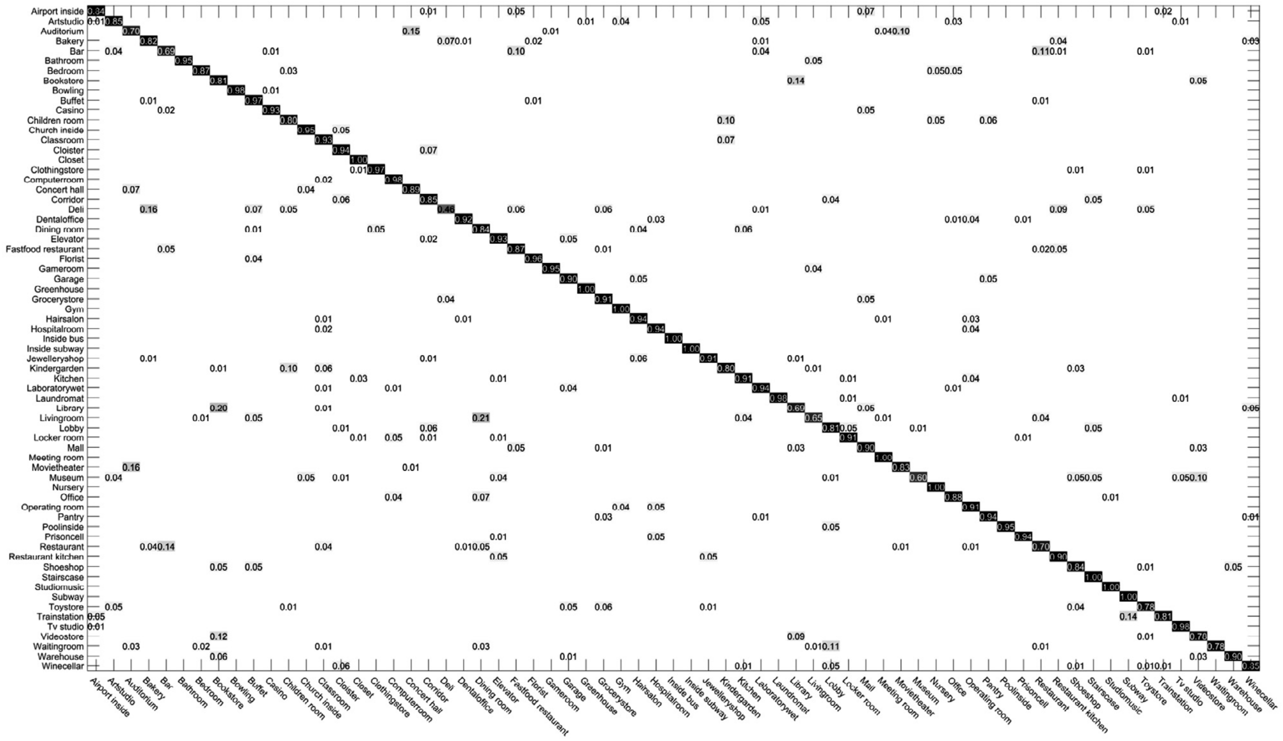


Fig. 8. Confusion matrix of DFF-ADML on the MIT-67 dataset.

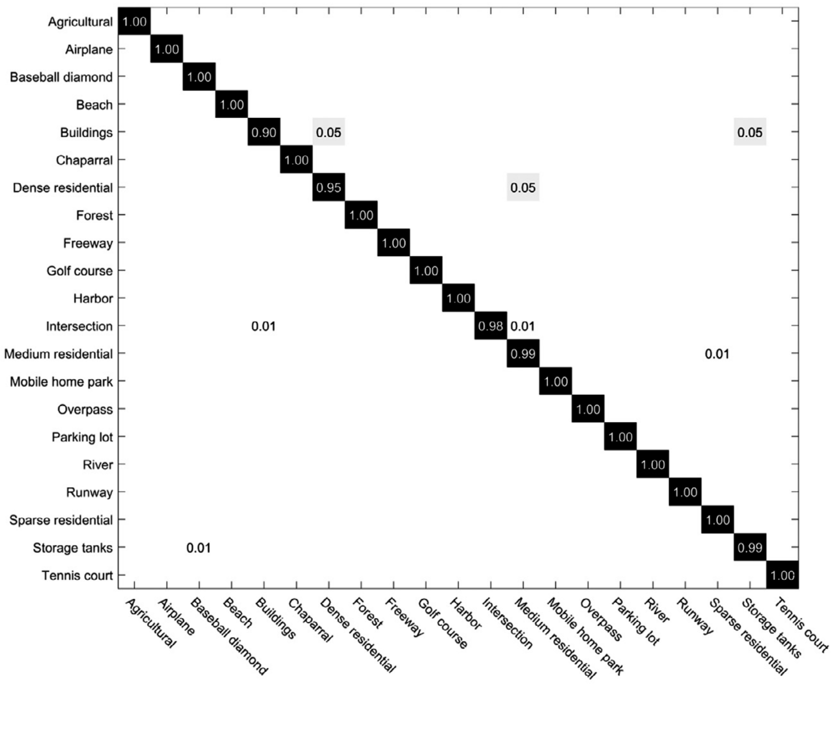


Fig. 9. Confusion matrix of DFF-ADML on the UCM-21 dataset.

5. Conclusion

In this paper, from the viewpoint of metric learning, we have proposed a novel deep feature fusion method for scene recognition. We have formulated an adaptive discriminative metric learning problem, which simultaneously exploits discriminative information from each deep feature vector and adaptively fuses complementary information from different deep feature vectors. Besides, we have mapped different deep feature vectors of the same image into a common space by different linear transformations, such that the consistent information can be preserved as much as possible. Extensive experiments on three benchmark scene datasets have demonstrated the superiority and robustness of the proposed deep feature fusion method. However, the performance of our kernelized version is not that impressive. In future work, we will explore more suitable kernel functions to further improve the recognition performance of the proposed method.

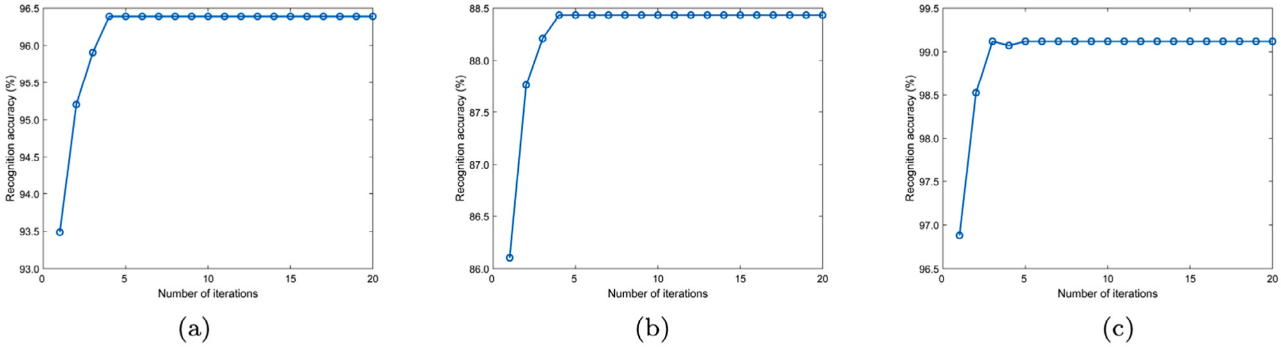


Fig. 10. Recognition accuracy of DFF-ADML versus different number of iterations on the three scene datasets. (a) Scene-15; (b) MIT-67; (c) UCM-21.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Chen Wang: Methodology, Software, Writing - original draft. **Guohua Peng:** Visualization, Supervision. **Bernard De Baets:** Validation, Supervision, Writing - review & editing.

Acknowledgment

This research received funding from the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme.

References

[1] W. Zhou, H. Li, J. Sun, Q. Tian, Collaborative index embedding for image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 40 (5) (2018) 1154–1166.

[2] M.A. Kaljahi, S. Palaiahnakote, M.H. Anisi, M.Y.I. Idris, M. Blumenstein, M.K. Khan, A scene image classification technique for a ubiquitous visual surveillance system, Multimed. Tool. Appl. 78 (5) (2019) 5791–5818.

[3] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1794–1801.

[4] J. Zou, W. Li, C. Chen, Q. Du, Scene classification using local and global features with collaborative representation fusion, Inf. Sci. (Ny) 348 (2016) 209–226.

[5] X. Cheng, J. Lu, J. Feng, Y. Bo, Z. Jie, Scene recognition with objectness, Pattern Recognit. 74 (2018) 474–487.

[6] P. Tang, H. Wang, S. Kwong, G-MS2F: GoogLeNet based multi-stage feature fusion of deep CNN for scene recognition, Neurocomputing 225 (2017) 188–197.

[7] L. Wang, S. Guo, W. Huang, Y. Xiong, Y. Qiao, Knowledge guided disambiguation for large-scale scene classification with multi-resolution CNNs, IEEE Trans. Image Process. 26 (4) (2017) 2055–2068.

[8] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, A. Oliva, Learning deep features for scene recognition using places database, in: Proceedings of the 27th International Conference on Neural Information Processing Systems, 2014, pp. 487–495.

[9] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: a 10 million image database for scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (6) (2018) 1452–1464.

[10] B. Liu, J. Liu, J. Wang, H. Lu, Learning a representative and discriminative part model with deep convolutional features for scene recognition, in: Proceedings of the Asian Conference on Computer Vision, 2014, pp. 643–658.

[11] G. Xie, X. Zhang, S. Yan, C. Liu, Hybrid CNN and dictionary-based models for scene recognition and domain adaptation, IEEE Trans. Circuits Syst. Video Technol. 27 (6) (2017) 1263–1274.

[12] X. Zhou, K. Jin, M. Xu, G. Guo, Learning deep compact similarity metric for kinship verification from face images, Inf. Fusion 48 (2019) 84–94.

[13] E.P. Xing, A.Y. Ng, M.I. Jordan, S. Russell, Distance metric learning, with application to clustering with side-information, in: Proceedings of the 15th International Conference on Neural Information Processing Systems, 2002, pp. 521–528.

[14] J.V. Davis, B. Kulis, P. Jain, S. Sra, I.S. Dhillon, Information-theoretic metric learning, in: Proceedings of the 24th International Conference on Machine Learning, 2007, pp. 209–216.

[15] S. Xiang, F. Nie, C. Zhang, Learning a Mahalanobis distance metric for data clustering and classification, Pattern Recognit. 41 (12) (2008) 3600–3612.

[16] J. Goldberger, S. Roweis, G. Hinton, R. Salakhutdinov, Neighbourhood components analysis, in: Proceedings of the 17th International Conference on Neural Information Processing Systems, 2004, pp. 513–520.

[17] K.Q. Weinberger, L.K. Saul, Distance metric learning for large margin nearest neighbor classification, J. Mach. Learn. Res. 10 (2009) 207–244.

[18] V.E. Liong, J. Lu, Y. Ge, Regularized local metric learning for person re-identification, Pattern Recognit. Lett. 68 (2015) 288–296.

[19] J. Zhang, X. Zhao, Integrated global-local metric learning for person re-identification, in: Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision, 2017, pp. 596–604.

[20] B. Nguyen, F.J. Ferri, C. Morell, B. De Baets, An efficient method for clustered multi-metric learning, Inf. Sci. (Ny) 471 (2019) 149–163.

[21] S.H. Khan, M. Hayat, M. Bennamoun, R. Togneri, F.A. Sohel, A discriminative representation of convolutional features for indoor scene recognition, IEEE Trans. Image Process. 25 (7) (2016) 3372–3383.

[22] L. Yang, X. Xie, P. Li, D. Zhang, L. Zhang, Part-based convolutional neural network for visual recognition, in: Proceedings of the 2017 IEEE International Conference on Image Processing, 2017, pp. 1772–1776.

[23] S. Guo, W. Huang, L. Wang, Y. Qiao, Locally-supervised deep hybrid model for scene recognition, IEEE Trans. Image Process. 26 (2) (2017) 808–820.

[24] L. Ye, W. Lei, Y. Sun, L. Zhao, Y. Wei, Parallel multi-stage features fusion of deep convolutional neural networks for aerial scene classification, Remote Sens. Lett. 9 (3) (2018) 294–303.

[25] Y. Yu, F. Liu, A two-stream deep fusion framework for high-resolution aerial scene classification, Comput. Intell. Neurosci. 2018 (2018) 1–13.

[26] N. Sun, W. Li, J. Liu, G. Han, C. Wu, Fusing object semantics and deep appearance features for scene recognition, IEEE Trans. Circuits. Syst. Video Technol. 29 (6) (2019) 1715–1728.

[27] T. Ojala, M. Pietikäinen, T. Mäenpää, Gray scale and rotation invariant texture classification with local binary patterns, in: Proceedings of the European Conference on Computer Vision, 2000, pp. 404–420.

[28] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.

[29] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Proceedings of the ECCV international Workshop on Statistical Learning in Computer Vision, 2004, pp. 1–22.

[30] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: Proceedings of the IEEE Computer Vision and Pattern Recognition, 2006, pp. 2169–2178.

[31] J. Sánchez, F. Perronnin, T. Mensink, J. Verbeek, Image classification with the fisher vector: theory and practice, Int. J. Comput. Vis. 105 (3) (2013) 222–245.

[32] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in: Proceedings of the 22nd ACM International Conference on Multimedia, 2014, pp. 675–678.

[33] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Proceedings of the 25th International Conference on Neural Information Processing Systems, 2012, pp. 1097–1105.

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[35] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of the International Conference on Learning Representations, 2015, pp. 1–14.

[36] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[37] S. Liu, G. Tian, Y. Xu, A novel scene classification model combining ResNet based transfer learning and data augmentation with a filter, Neurocomputing 338 (2019) 191–206.

[38] B. Nguyen, C. Morell, B. De Baets, Large-scale distance metric learning for *k*-nearest neighbors regression, Neurocomputing 214 (2016) 805–814.

[39] C. Lopez-Molina, J. Montero, H. Bustince, B. De Baets, Self-adapting weighted operators for multiscale gradient fusion, Inf. Fusion 44 (2018) 136–146.

[40] M. Wang, X. Hua, X. Yuan, Y. Song, L. Dai, Optimizing multi-graph learning: towards

a unified video annotation scheme, in: Proceedings of the 15th ACM International Conference on Multimedia, 2007, pp. 862–871.

[41] B. Nguyen, B. De Baets, Kernel distance metric learning using pairwise constraints for person re-identification, IEEE Trans. Image Process. 28 (2) (2019) 589–600.

[42] J. Wang, H. Do, A. Woznica, A. Kalousis, Metric learning with multiple kernels, in: Proceedings of the 24th International Conference on Neural Information Processing Systems, 2011, pp. 1170–1178.

[43] A. Quattoni, A. Torralba, Recognizing indoor scenes, in: Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 413–420.

[44] Y. Yang, S. Newsam, Bag-of-visual-words and spatial extensions for land-use classification, in: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2010, pp. 270–279.

[45] J. Deng, W. Dong, R. Socher, L. Li, K. Li, Fei-Fei. Li, ImageNet: a large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[46] J. Yu, D. Tao, Y. Rui, J. Cheng, Pairwise constraints based multiview features fusion for scene classification, Pattern Recognit. 46 (2) (2013) 483–496.

[47] L.J. Li, H. Su, Fei-Fei. Li, E.P. Xing, Object banks: a high-level image representation for scene classification and semantic feature sparsification, in: Advances in Neural Information Processing Systems, 2010, pp. 1378–1386.

[48] L. Xie, F. Lee, L. Liu, Z. Yin, Y. Yan, W. Wang, J. Zhao, Q. Chen, Improved spatial pyramid matching for scene recognition, Pattern Recognit. 82 (2018) 118–129.

[49] Z. Zuo, G. Wang, B. Shuai, L. Zhao, Q. Yang, Exemplar based deep discriminative and shareable feature learning for scene image classification, Pattern Recognit. 48 (10) (2015) 3004–3015.

[50] Z. Zuo, G. Wang, B. Shuai, L. Zhao, Q. Yang, X. Jiang, Learning discriminative and shareable features for scene classification, in: European Conference on Computer Vision, 2014, pp. 552–568.

[51] R. Margolin, L. Zelnik-Manor, A. Tal, OTC: a novel local descriptor for scene classification, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 377–391.

[52] M. Juneja, A. Vedaldi, C.V. Jawahar, A. Zisserman, Blocks that shout: distinctive parts for scene classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 923–930.

[53] G.S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, AID: a benchmark dataset for performance evaluation of aerial scene classification, IEEE Trans. Geosci. Remote Sens. 55 (7) (2017) 3965–3981.

[54] E. Flores, M. Zortea, J. Scharcanski, Dictionaries of deep features for land-use scene classification of very high spatial resolution images, Pattern Recognit. 89 (2019) 32–44.