Discriminative Analysis Dictionary Learning with Adaptively Ordinal Locality Preserving

Jing Dong[1,*], Kai Wu[1], Chang Liu[2], Xue Mei[1], Wenwu Wang[3]

Abstract

Dictionary learning has found broad applications in signal and image processing. By adding constraints to the traditional dictionary learning model, dictionaries with discriminative capability can be obtained which can deal with the task of image classification. The Discriminative Convolutional Analysis Dictionary Learning (DCADL) algorithm proposed recently has achieved promising results with low computational complexity. However, DCADL is still limited in classification performance because of the lack of constraints on dictionary structures. To solve this problem, this study introduces an adaptively ordinal locality preserving (AOLP) term to the original model of DCADL to further improve the classification performance. With the AOLP term, the distance ranking in the neighborhood of each atom can be preserved, which can improve the discrimination of coding coefficients. In addition, a linear classifier for the classification of coding coefficients is trained along with the dictionary. A new method is designed specifically to solve the optimization problem corresponding to the proposed model. Experiments are performed on several commonly used datasets to show the promising results of the proposed algorithm in classification performance and computational efficiency.

Keywords: Discriminative dictionary learning, Ordinal locality preserving,

*Corresponding author.

[1]J. Dong, K. Wu, and X. Mei are with the College of Electrical Engineering and Control Science, Nanjing Tech University, Nanjing, Jiangsu, China. (email: jingdong@njtech.edu.cn)

[2]C. Liu is with Qingdao Hatran Ocean Intelligence Technology Co., Ltd, Qingdao, Shandong, China.

[3]W. Wang is with the Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, GU2 7XH, UK.

1. Introduction

Sparse representation and dictionary learning have been successfully applied to various tasks in signal processing and computer vision, e.g. signal declipping [1], image super-resolution [2], [3], image denoising [4], [5], [6], and pattern recognition [7], [8]. In sparse representation, the signals of interest are represented using a small number of atoms, i.e., signal components, chosen from a dictionary. The objective of dictionary learning is to determine the dictionary based on a set of training data. As dictionaries learned from training examples tend to represent signals more accurately as compared with pre-defined dictionaries [9], [10], dictionary learning has gained much attention in the past decade.

A well-known dictionary learning method is the synthesis model based dictionary learning (SDL), which learns an over-complete synthesis dictionary so that signals can be decomposed as linear combinations of dictionary atoms. The SDL model is originally proposed for signal recovery [9], [11], and it has been adapted to classification and recognition tasks by considering labels of training samples and imposing additional constraints to improve the ability of the learned dictionaries in discriminating classes. For example, Ramirez et al. [12] and Sun et al. [13] both added an incoherence promoting term to the traditional SDL model to improve the incoherence between dictionaries corresponding to different classes. In addition, Sun et al. [13] imposed a discriminative fidelity term to encourage each sub-dictionary associated with a specific category to sparsify the samples in the corresponding class. In [14], a new label consistency constraint is proposed to guarantee the discrimination of sparse coefficients during the dictionary learning process. These methods have obtained encouraging classification results due to the employment of additional constraints. However, as the estimation of sparse coefficients in SDL has relatively high computational complexity, the efficiency of the SDL-based classification methods needs to be

further improved.

Different from SDL, the model of analysis dictionary learning (ADL) represents signals as sparse vectors by multiplying an analysis dictionary matrix with the vectors representing the signals. ADL has been widely used in image and signal reconstruction [15], [10], [1], however, its applications in classification have not attracted much attention due to the relatively limited discrimination capability of the original model. The computational efficiency of ADL in sparse coding may help reduce the time complexity of dictionary-learning-based classification methods, it is worth studying the potential of discriminative ADL for classification problems. Tang et al. [16] imposed structural information on ADL and developed a structural analysis dictionary learning (SADL) approach. To integrate the advantages of ADL and SDL, a novel approach named projective dictionary pair learning (DPL) [17] was proposed, which attempts to learn an analysis dictionary and a synthesis dictionary simultaneously with improved discrimination ability and computational efficiency. Some methods have been developed to further improve the original DPL algorithm. For example, a discriminative DPL algorithm [18] was developed by learning a linear classifier together with the dictionary. Chen et al. [19] learned relaxed block-diagonal representations of signals by imposing a locality constraint and proposed a structured version of DPL named as relaxed block-diagonal DPL (RBD-DPL). In [20], a low-rank analysis-synthesis dictionary learning (LR-ASDL) approach was proposed, where a regularizer based on the rank of the synthesis dictionary is applied and an adaptively ordinal locality preserving (AOLP) term was developed to preserve the distance ranking of dictionary atoms.

Different from the algorithms mentioned above, Tang et al. [21] regarded atoms of an analysis dictionary as convolutional filters and proposed a Discriminative Convolutional Analysis Dictionary Learning (DCADL) algorithm. This algorithm learns the analysis dictionary which leads to sparse feature maps for signals. A linear classifier based on the feature maps is also trained simultaneously. To improve the efficiency of optimization, DCADL reshapes the input signal to multiple segments each having same dimension as the dictionary atom

and then converts the convolution operation in the original model to matrix multiplication. This algorithm achieves competitive results for image classification with low training and testing time costs.

Although DCADL [21] achieves promising results while greatly reducing the computational complexity, the lack of consideration on the dictionary structure information limits its classification performance. As the employment of discrimination constraints has been shown effective in increasing the classification accuracies obtained by dictionary learning algorithms, complementing the lack of additional constraints in the model of DCADL may improve its discrimination capability and classification performance. Therefore, in this study we propose to introduce structural constraints on dictionary atoms to the model of DCADL to further improve its discrimination capability.

In particular, it has been shown in LR-ASDL [20] that the AOLP term is effective in preserving the locally structural information of the learned dictionary. Inspired by this idea, we apply the AOLP term to complement the lack of structure constraints for the dictionary in DCADL. In this way, the discriminative capability of the dictionary and sparse coefficients could be improved, which can potentially improve the classification performance of the trained classifier. The multi-variable optimization problem formulated from this model is solved using an alternating optimization framework. Due to the fast coding capability of ADL and enhanced intra-class compactness provided by the AOLP term, the proposed algorithm can obtain higher classification accuracies than DCADL with relatively low time complexity. Experiments on standard databases, including Extended YaleB [22], UCF-50 [23], Caltech101 [24] and Scene15 [25], show that the proposed model generally performs better than several classical and state-of-the-art methods, e.g., DPL [17], SADL [16] and RBD-DPL [19].

The remaining sections are organized as follows. Section 2 presents some related work including the general model for discriminative ADL, the DCADL algorithm and the AOLP term. Section 3 introduces the proposed formulation and the corresponding optimization framework. Section 4 analyzes the time complexity of the proposed algorithm. Section 5 presents experimental results

as compared with other algorithms and Section 6 draws the main conclusions of the study.

2. Related Work

2.1. Discriminative ADL

To learn an analysis dictionary leading to sparse coefficients with discriminative ability, discriminative ADL usually adds additional discrimination terms or constraints to the model of traditional ADL. Let $\mathbf{X}$ denote the training signals and $\mathbf{Y}$ denote the corresponding labels. Discriminative ADL is generally formulated as

$$\arg\min_{\mathbf{\Omega},\mathbf{U}} \frac{1}{2}\|\mathbf{\Omega X}-\mathbf{U}\|_F^2+\lambda\|\mathbf{U}\|_p+f(\mathbf{\Omega},\mathbf{U},\mathbf{Y}), \tag{2.1}$$

where the first term $\|\mathbf{\Omega X}-\mathbf{U}\|_F^2$ models the reconstruction error obtained using the learned analysis dictionary $\mathbf{\Omega}$, and $\|\cdot\|_F$ denotes the Frobenius norm. $\|\cdot\|_p$ denotes the ℓ_p-norm ($0<p\leq 1$), and $\|\mathbf{U}\|_p$ promotes the sparsity of the coefficient matrix $\mathbf{U}$, and λ represents the regularization parameter. The last term $f(\mathbf{\Omega},\mathbf{U},\mathbf{Y})$ denotes the constraint function to improve the discriminative capability of $\mathbf{\Omega}$ and $\mathbf{U}$ by considering the label information $\mathbf{Y}$. For example, in [14], a label consistency matrix and a linear transformation of the coefficient matrix are employed as the constraint function. In [20], a structured discriminant term is used as the constraint function, which can enhance the discriminative ability of analysis sub-dictionaries. Discriminative ADL algorithms have been widely used to deal with tasks in pattern recognition/classification, such as target recognition in synthetic aperture radar (SAR) images [26], facial expression recognition [27], and physical activity recognition [28].

2.2. DCADL

Tang et al. [21] introduced a convolutional mapping to the framework of the conventional discriminative ADL and proposed the DCADL model as follows

$$\arg \min_{\boldsymbol{\omega}_i, \boldsymbol{u}_j^i, \mathbf{W}} \sum_{j=1}^{n} \sum_{i=1}^{m} \left(\frac{1}{2} \left\| \boldsymbol{\omega}_i * \boldsymbol{x}_j - \boldsymbol{u}_j^i \right\|_2^2 + \lambda_1 \left\| \boldsymbol{u}_j^i \right\|_1 \right) + \frac{\lambda_2}{2} \|\mathbf{Y} - \mathbf{W}\widetilde{\mathbf{U}}\|_F^2,$$
$$\text{s.t. } \|\boldsymbol{\omega}_i\|_2^2 \leq 1, \forall i = 1, \ldots, m; \ \widetilde{\mathbf{U}} = \begin{bmatrix} \boldsymbol{u}_1^1 & \cdots & \boldsymbol{u}_n^1 \\ \vdots & \ddots & \vdots \\ \boldsymbol{u}_1^m & \cdots & \boldsymbol{u}_n^m \end{bmatrix}, \tag{2.2}$$

where $*$ denotes the convolution operator, $\boldsymbol{\omega}_i^T \in \mathbb{R}^{s^2}$ represents the i-th atom (row) of the analysis dictionary, $\boldsymbol{x}_j$ denotes the j-th training sample, $\boldsymbol{u}_j^i$ denotes the i-th response map of $\boldsymbol{x}_j$ corresponding to the convolution of $\boldsymbol{\omega}_i$ and $\boldsymbol{x}_j$, and $\mathbf{W}$ denotes a linear classifier to be learned. λ_1 and λ_2 are penalty parameters. This framework considers the atoms of the analysis dictionary as linear filters and assumes the response maps of signals to be approximately sparse. The linear classifier $\mathbf{W}$ is jointly learned based on the response maps and label information.

It should be noted that DCADL takes dictionary atoms as filters and obtains the response maps of signals via convolution operations, rather than employing the matrix multiplication of the dictionary and signal matrices as in traditional ADL models. In this way, the dictionary can learn global features of input signals, rather than redundant local features like the traditional dictionary. To solve the obtained optimization problem, the original formulation is converted to a simplified form without the convolution operation by reshaping the data matrix and coefficient matrix [21]. Matrix reshaping operators are also employed to update related variables in the optimization procedure. This formulation and the corresponding optimization method have been shown effective for classification tasks with low time complexity.

2.3. Adaptively Ordinal Locality Preserving

The AOLP term proposed in [20] is based on the work of ordinal locality preserving (OLP) [29] for feature selection, where an OLP loss function is developed to maintain the ordinal locality of original data in selected features. Related works on locality preserving can be also found in works for clustering [30], [31], [32], where local manifold information is preserved using local cen-

troids or graph regularization. In particular, the OLP in feature selection is constructed as follows [29].

Definition 1. *Given a triplet* $(\boldsymbol{y}_i, \boldsymbol{y}_u, \boldsymbol{y}_v)$ *with* $\boldsymbol{y}_u$ *and* $\boldsymbol{y}_v$ *being the neighbors of* $\boldsymbol{y}_i$*, let* $(\boldsymbol{f}_i, \boldsymbol{f}_u, \boldsymbol{f}_v)$ *denote their corresponding features. The feature selection process is considered to be ordinal locality preserving when the following condition is satisfied: if* $\operatorname{dist}(\boldsymbol{y}_i, \boldsymbol{y}_u) \leq \operatorname{dist}(\boldsymbol{y}_i, \boldsymbol{y}_v)$*, then* $\operatorname{dist}(\boldsymbol{f}_i, \boldsymbol{f}_u) \leq \operatorname{dist}(\boldsymbol{f}_i, \boldsymbol{f}_v)$*, where* $\operatorname{dist}(.,.)$ *denotes a distance metric.*

Based on the above definition, the OLP loss function is defined as

$$\max_{\boldsymbol{F}} \sum_{i=1}^{n} \sum_{u \in \mathcal{N}_i} \sum_{v \in \mathcal{N}_i} S_{uv}^{i} \left[\operatorname{dist}(\boldsymbol{f}_i, \boldsymbol{f}_u) - \operatorname{dist}(\boldsymbol{f}_i, \boldsymbol{f}_v) \right], \tag{2.3}$$

where $\boldsymbol{F} = \{\boldsymbol{f}_1, \boldsymbol{f}_2 ..., \boldsymbol{f}_n\}$ is the set consisting of all feature vectors, the set $\mathcal{N}_i$ contains the indices of k nearest neighbors of $\boldsymbol{y}_i$, and $\boldsymbol{S}^i$ is an anti-symmetric matrix, where its (u, v)-th element is $S_{uv}^{i} = \operatorname{dist}(\boldsymbol{y}_i, \boldsymbol{y}_u) - \operatorname{dist}(\boldsymbol{y}_i, \boldsymbol{y}_v)$.

As $\boldsymbol{S}^i$ is an antisymmetric matrix, we have $S_{uv}^{i} = -S_{vu}^{i}$. Based on this, the original loss function of OLP, i.e., equation (2.3), can be rewritten as

$$\max_{\boldsymbol{F}} \left(-\sum_{i=1}^{n} \sum_{u \in \mathcal{N}_i} \sum_{v \in \mathcal{N}_i} S_{vu}^{i} \operatorname{dist}(\boldsymbol{f}_i, \boldsymbol{f}_u) - \sum_{i=1}^{n} \sum_{v \in \mathcal{N}_i} \sum_{u \in \mathcal{N}_i} S_{uv}^{i} \operatorname{dist}(\boldsymbol{f}_i, \boldsymbol{f}_v) \right) \tag{2.4}$$

Inspired by [33], we define a weighting matrix $\boldsymbol{C} \in \mathbb{R}^{n \times n}$ whose (i, j)th element is defined as

$$C_{ij} = \begin{cases} \sum_{u \in \mathcal{N}_i} S_{uj}^{i}, & j \in \mathcal{N}_i, \\ 0, & j \notin \mathcal{N}_i. \end{cases} \tag{2.5}$$

By substituting the weighting matrix $\boldsymbol{C}$ to (2.4), the loss function of OLP is rewritten as

$$\max_{\boldsymbol{F}} \left(-\sum_{i=1}^{n} \sum_{u=1}^{n} C_{iu} \operatorname{dist}(\boldsymbol{f}_i, \boldsymbol{f}_u) - \sum_{i=1}^{n} \sum_{v=1}^{n} C_{iv} \operatorname{dist}(\boldsymbol{f}_i, \boldsymbol{f}_v) \right), \tag{2.6}$$

which is equivalent to [29]

$$\min_{\boldsymbol{F}} \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} \operatorname{dist}(\boldsymbol{f}_i, \boldsymbol{f}_j). \tag{2.7}$$

In [20], Li et al. adapted the OLP concept [29] to dictionary learning, and proposed adaptively ordinal locality preserving (AOLP). In particular, they have proved that if analysis dictionary atoms $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_j$ are similar enough, their corresponding profiles $\boldsymbol{p}_i = \boldsymbol{\omega}_i\mathbf{X}$ and $\boldsymbol{p}_j = \boldsymbol{\omega}_j\mathbf{X}$, i.e., rows of the coefficient matrix, should be similar. Based on this, the definition of OLP can be adapted to maintain the local structural information of the learned dictionary. As both the dictionary and profiles are updated in the learning process, the OLP term is also updated adaptively. Specifically, AOLP in ADL has the following definition [20].

Definition 2. *For a triplet of analysis dictionary atoms* $(\boldsymbol{w}_i, \boldsymbol{w}_u, \boldsymbol{w}_v)$ *with* $\boldsymbol{w}_u$ *and* $\boldsymbol{w}_v$ *being the neighbors of* $\boldsymbol{w}_i$*, let* $(\boldsymbol{p}_i, \boldsymbol{p}_u, \boldsymbol{p}_v)$ *denote the corresponding profiles of data* $\mathbf{X}$*, i.e.,* $\boldsymbol{p}_i = \boldsymbol{w}_i\mathbf{X}$*,* $\boldsymbol{p}_u = \boldsymbol{w}_u\mathbf{X}$*, and* $\boldsymbol{p}_v = \boldsymbol{w}_v\mathbf{X}$*. The analysis dictionary learning process is considered to be adaptively ordinal locality preserving when the following condition is satisfied: if* $\operatorname{dist}(\boldsymbol{w}_i, \boldsymbol{w}_u) \leq \operatorname{dist}(\boldsymbol{w}_i, \boldsymbol{w}_v)$*, then* $\operatorname{dist}(\boldsymbol{p}_i, \boldsymbol{p}_u) \leq \operatorname{dist}(\boldsymbol{p}_i, \boldsymbol{p}_v)$*, where* $\operatorname{dist}(.,.)$ *denotes a distance metric.*

According to Definitions 1 and 2, AOLP can be developed by adapting the loss function of OLP, i.e., equations (2.5) and (2.7). Specifically, the loss function of AOLP can be formulated as [20]

$$\sum_{i=1}^{m}\sum_{v=1}^{m} B_{iv} \operatorname{dist}(\boldsymbol{p}_i, \boldsymbol{p}_v), \tag{2.8}$$

where

$$B_{iv} = \begin{cases} \sum_{u\in\mathcal{Z}_i} V_{uv}^i, & v \in \mathcal{Z}_i, \\ 0, & v \notin \mathcal{Z}_i, \end{cases} \tag{2.9}$$

and $\mathcal{Z}_i$ denotes a list indicating the k nearest atoms of the i-th atom $\boldsymbol{w}_i$. The (u, v)-th element of the antisymmetric matrix $\boldsymbol{V}^i$ is defined as

$$V_{uv}^i = \operatorname{dist}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_u) - \operatorname{dist}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_v). \tag{2.10}$$

The employment of AOLP in ADL can be seen as applying additional constraints to the dictionary and the coefficient matrix, where the update of these

two variables is enforced to be consistent in terms of ordinal locality. The LR-ASDL algorithm proposed in [20] employs the AOLP term in dictionary learning, which has improved the discrimination capability of the obtained dictionary. However, LR-ASDL has high computational complexity and is time-consuming to run in the training stage, as an analysis dictionary and a low-rank synthesis dictionary are learned simultaneously.

DCADL [21] achieves promising performance on classification tasks with relatively low time complexity. However, local structures of the analysis dictionary are not considered in DCADL, which may limit the accuracies of classification. AOLP [20] can maintain the ordinal locality of the dictionary which is helpful to improve the discrimination ability. Therefore, such a term could also be used with DCADL to exploit the dictionary structure information while maintaining a low level of computational complexity, as shown in our work discussed next.

3. Proposed Algorithm

The proposed model is introduced first and then converted to an optimization problem that is easier to solve. After that, the detailed procedure of optimization is presented.

3.1. Proposed formulation

We introduce the AOLP term to the formulation of DCADL, and propose the following formulation:

$$\begin{aligned}\arg \min_{\boldsymbol{\omega}_i, \boldsymbol{u}_j^i, \mathbf{W}} & \sum_{j=1}^{n}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|\boldsymbol{\omega}_i * \boldsymbol{x}_j - \boldsymbol{u}_j^i\right\|_2^2 + \lambda_1\left\|\boldsymbol{u}_j^i\right\|_1\right) + \mu\sum_{i=1}^{m}\sum_{v=1}^{m} B_{iv}\,\mathrm{dist}\left(\boldsymbol{p}_i, \boldsymbol{p}_v\right) \\ & + \frac{\lambda_2}{2}\|\mathbf{Y} - \mathbf{W}\widetilde{\mathbf{U}}\|_F^2 + \frac{\lambda_3}{2}\|\mathbf{W}\|_F^2 + \frac{\lambda_4}{2}\|\boldsymbol{\Omega}\|_F^2, \\ & \text{s.t. } \|\boldsymbol{\omega}_i\|_2^2 \le 1, \forall i = 1, \ldots, m;\ \widetilde{\mathbf{U}} = \begin{bmatrix} \boldsymbol{u}_1^1 & \cdots & \boldsymbol{u}_n^1 \\ \vdots & \ddots & \vdots \\ \boldsymbol{u}_1^m & \cdots & \boldsymbol{u}_n^m \end{bmatrix},\end{aligned} \tag{3.1}$$

where dist $(.,.)$ denotes the squared Euclidean distance. The vector $\boldsymbol{\omega}_i^T \in \mathbb{R}^{s^2}$ is the i-th atom/row of $\boldsymbol{\Omega}$, and $\boldsymbol{x}_j \in \mathbb{R}^r$ denotes the j-th image. The vector $\boldsymbol{u}_j^i \in \mathbb{R}^p$ denotes the i-th feature map of the j-th image extracted using the i-th atom, and $\boldsymbol{p}^i = \boldsymbol{w}_i \mathbf{X}$ is the i-th profile of data $\mathbf{X}$ corresponding to the atom $\boldsymbol{w}_i$. Matrix $\mathbf{Y} \in \mathbb{R}^{c \times n}$ denotes the label matrix, and $\mathbf{W} \in \mathbb{R}^{c \times mp}$ denotes the linear classifier to be learned simultaneously. The definition of the AOLP term $\sum_{i=1}^{m} \sum_{v=1}^{m} B_{iv} \operatorname{dist}(\boldsymbol{p}_i, \boldsymbol{p}_v)$ is based on equations (2.9)-(2.10). The regularization parameters $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ and μ are chosen empirically. Detailed discussions about parameter selection are given in Section 5.3.

In the proposed formulation, each atom of the dictionary acts as a convolutional kernel, and the response maps are assumed to be sparse, which correspond to the first two terms in equation (3.1). As compared with the formulation of DCADL, i.e., equation (2.2), three terms have been introduced in the proposed formulation (3.1), namely, the AOLP term, $\|\boldsymbol{\Omega}\|_F^2$ and $\|\mathbf{W}\|_F^2$. The AOLP term is to improve the consistency between the ordinal locality of dictionary atoms and that of coefficient profiles. The terms $\|\boldsymbol{\Omega}\|_F^2$ and $\|\mathbf{W}\|_F^2$ can limit the values of $\mathbf{W}$ and $\boldsymbol{\Omega}$, and enhance the generalization ability of the model via regularizations [34]. As compared with LR-ASDL [20] which also employs the AOLP term, the proposed formulation only learns an analysis dictionary while the formulation of LR-ASDL attempts to learn an analysis dictionary and a low-rank synthesis dictionary simultaneously. The proposed algorithm is named as Discriminative Analysis Dictionary Learning with AOLP (DADL-AOLP).

3.2. Optimization strategy

The proposed formulation (3.1) is reformulated by simplifying the objective function which is then addressed with an optimization method.

3.2.1. Simplified objective function

To address the proposed formulation (3.1) efficiently, we reformulate the problem by converting the convolution calculation and the AOLP term into matrix operations. Each image is segmented into patches of size $s \times s$,

which is the same as the size of dictionary atoms. Let $\mathbf{x}_{i_p} \in \mathbb{R}^{s^2}$ denote the p-th patch of the i-th sample, and all image patches can be denoted by $\mathbf{X} = \left[\mathbf{x}_{1_1}, \ldots, \mathbf{x}_{1_p}, \ldots, \mathbf{x}_{n_1}, \ldots, \mathbf{x}_{n_p}\right] \in \mathbb{R}^{s^2 \times np}$. The feature maps of all image patches are denoted by

$$\overline{\mathbf{U}} = \begin{bmatrix} u_{1_1}^1 & \cdots & u_{1_p}^1 & \cdots & u_{n_1}^1 & \cdots & u_{n_p}^1 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ u_{1_1}^m & \cdots & u_{1_p}^m & \cdots & u_{n_1}^m & \cdots & u_{n_p}^m \end{bmatrix} \in \mathbb{R}^{m \times np}. \tag{3.2}$$

Note that $\overline{\mathbf{U}} \in \mathbb{R}^{m \times np}$ and $\widetilde{\mathbf{U}} \in \mathbb{R}^{mp \times n}$ are composed of the same elements but with different arrangements, and the transformation between them can be realized using the matrix reshaping operator defined as follows.

Definition 3. *Assuming a matrix $\overline{\mathbf{U}} \in \mathbb{R}^{m \times np}$ is represented as a block matrix by grouping every p columns as a block, that is $\overline{\mathbf{U}} = [\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_n]$ where $\mathbf{U}_i \in \mathbb{R}^{m \times p}$ for $i = 1, \ldots, n$. By stacking the columns of $\mathbf{U}_i$ as one column vector $\mathbf{u}_i \in \mathbb{R}^{mp}$, $\overline{\mathbf{U}}$ can be reshaped to a matrix $\widetilde{\mathbf{U}} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n] \in \mathbb{R}^{mp \times n}$. The matrix reshaping process from $\overline{\mathbf{U}}$ to $\widetilde{\mathbf{U}}$ is defined as an operator $\mathcal{R}$, and the inverse process is defined as $\mathcal{R}^{-1}$, i.e., $\widetilde{\mathbf{U}} = \mathcal{R}(\overline{\mathbf{U}})$, and $\overline{\mathbf{U}} = \mathcal{R}^{-1}(\widetilde{\mathbf{U}})$.*

Inspired by [20], with some computation, the AOLP term

$$\sum_{i=1}^{m} \sum_{v=1}^{m} B_{iv} \operatorname{dist}\left(\boldsymbol{p}_i, \boldsymbol{p}_v\right) \tag{3.3}$$

can be converted to

$$\operatorname{Tr}\left(\overline{\mathbf{U}}^T \mathbf{L} \overline{\mathbf{U}}\right), \tag{3.4}$$

where $\mathbf{L}$ denotes the Laplacian matrix defined by

$$\mathbf{L} \triangleq \mathbf{M} - \frac{\mathbf{B} + \mathbf{B}^T}{2}, \tag{3.5}$$

and the (i, i)-th element of $\mathbf{M}$ is

$$M_{ii} = \sum_{v=1}^{m} \frac{B_{iv} + B_{vi}}{2} \tag{3.6}$$

Thus, the original model (3.1) is reformulated as

$$\begin{aligned}
\arg \min_{\substack{\boldsymbol{\Omega},\overline{\mathbf{U}},\widetilde{\mathbf{U}} \\ \mathbf{W},\mathbf{L}}} & \frac{1}{2}\|\boldsymbol{\Omega}\mathbf{X}-\overline{\mathbf{U}}\|_F^2+\lambda_1\|\widetilde{\mathbf{U}}\|_1+\frac{\lambda_2}{2}\|\mathbf{Y}-\mathbf{W}\widetilde{\mathbf{U}}\|_F^2+\mu\,\mathrm{tr}\left(\overline{\mathbf{U}}^T\mathbf{L}\overline{\mathbf{U}}\right) \\
& +\frac{\lambda_3}{2}\|\mathbf{W}\|_F^2+\frac{\lambda_4}{2}\|\boldsymbol{\Omega}\|_F^2 \\
\text{s.t. } \widetilde{\mathbf{U}} = & \begin{bmatrix} u_{1_1}^1 & \cdots & u_{n_1}^1 \\ \vdots & \ddots & \vdots \\ u_{1_p}^m & \cdots & u_{n_p}^m \end{bmatrix}, \overline{\mathbf{U}} = \begin{bmatrix} u_{1_1}^1 & \cdots & u_{1_p}^1 & \cdots & u_{n_1}^1 & \cdots & u_{n_p}^1 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ u_{1_1}^m & \cdots & u_{1_p}^m & \cdots & u_{n_1}^m & \cdots & u_{n_p}^m \end{bmatrix}, \\
& \|\boldsymbol{\omega}_i\|_2^2 \leq 1, \forall i = 1,\ldots,m,
\end{aligned} \tag{3.7}$$

where $\boldsymbol{\Omega} \in \mathbb{R}^{m\times s^2}$ denotes the analysis dictionary. Note that both $\widetilde{\mathbf{U}} \in \mathbb{R}^{mp\times n}$ and $\overline{\mathbf{U}} \in \mathbb{R}^{m\times np}$ contain the feature maps of all image patches, and $\widetilde{\mathbf{U}}$ is obtained by reshaping $\overline{\mathbf{U}}$ as given in Definition 3.

3.2.2. Optimization method

To address the reformulated problem (3.7), we alternatively update each variable while keeping the remaining variables fixed. As a result, the problem is converted to five subproblems and each subproblem is addressed alternatively. The analysis dictionary $\boldsymbol{\Omega}$ is randomly initialized and the detailed optimization steps in each iteration are as follows.

(1) Update Laplacian matrix $\mathbf{L}$:

Given an analysis dictionary $\boldsymbol{\Omega}$, the indices of k nearest neighbors of each atom $\boldsymbol{w}_i$, i.e., $\mathcal{Z}_i$, are decided by computing the squared Euclidean distances between $\boldsymbol{w}_i$ and the other atoms with $k = 2$. The matrix $\mathbf{B}$ can be updated based on equations (2.9) and (2.10), and then the Laplacian matrix $\mathbf{L}$ can be updated using equations (3.5) and (3.6).

(2) Update coefficients $\overline{\mathbf{U}}$:

Ignoring other unrelated variables, the cost function is simplified as

$$\arg \min_{\overline{\mathbf{U}}} \frac{1}{2}\|\boldsymbol{\Omega}\mathbf{X}-\overline{\mathbf{U}}\|_F^2+\mu\,\mathrm{tr}\left(\overline{\mathbf{U}}^T\mathbf{L}\overline{\mathbf{U}}\right). \tag{3.8}$$

The optimal solution to this subproblem can be obtained by calculating the gradient of the objective function and setting it to zero, i.e.,

$$\overline{\mathbf{U}} - \mathbf{\Omega}\mathbf{X} + 2\mu\mathbf{L}\overline{\mathbf{U}} = 0, \tag{3.9}$$

which gives the solution

$$\overline{\mathbf{U}} = (\mathbf{I} + 2\mu\mathbf{L})^{-1}\mathbf{\Omega}\mathbf{X}, \tag{3.10}$$

where $\mathbf{I}$ denotes the identity matrix and $(\cdot)^{-1}$ returns the inverse of a matrix.

(3) Update coefficients $\widetilde{\mathbf{U}}$:

$\widetilde{\mathbf{U}}$ can be updated based on the update of $\overline{\mathbf{U}}$ using the matrix reshaping operator $\mathcal{R}$ defined in Definition 3, i.e., $\widetilde{\mathbf{U}} = \mathcal{R}(\overline{\mathbf{U}})$. By removing other unrelated variables in (3.7), $\widetilde{\mathbf{U}}$ is then further optimized by addressing the subproblem as follows

$$\arg\ \min_{\widetilde{\mathbf{U}}} \frac{\lambda_2}{2}\|\mathbf{Y} - \mathbf{W}\widetilde{\mathbf{U}}\|_F^2 + \lambda_1\|\widetilde{\mathbf{U}}\|_1. \tag{3.11}$$

Let $f(\widetilde{\mathbf{U}}) = \frac{\lambda_2}{2}\|\mathbf{Y} - \mathbf{W}\widetilde{\mathbf{U}}\|_F^2$. According to [35], the solution can be obtained using an iterative shrinkage-thresholding algorithm, that is

$$\begin{aligned} \widetilde{\mathbf{U}} &= \text{prox}_{\rho\lambda_1}(\widetilde{\mathbf{U}} - \rho\nabla f(\widetilde{\mathbf{U}})) \\ &= \mathcal{T}_{\rho\lambda_1}\left(\widetilde{\mathbf{U}} - \rho\lambda_2\mathbf{W}^T\left(\mathbf{W}\widetilde{\mathbf{U}} - \mathbf{Y}\right)\right), \end{aligned} \tag{3.12}$$

where ρ is a step size, and $\mathcal{T}_\alpha(\cdot)$ is the shrinkage operator defined by

$$\mathcal{T}_\alpha(x) = (|x| - \alpha)_+ \,\text{sgn}\,(x) \tag{3.13}$$

where α is the soft-threshold, and $\text{sgn}(x)$ returns the sign of x.

(4) Update classifier $\mathbf{W}$:

Ignoring the other variables, $\mathbf{W}$ is optimized by solving the optimization problem

$$\arg\ \min_{\mathbf{W}} \frac{\lambda_2}{2}\|\mathbf{Y} - \mathbf{W}\widetilde{\mathbf{U}}\|_F^2 + \frac{\lambda_3}{2}\|\mathbf{W}\|_2^2. \tag{3.14}$$

Set the gradient of (3.14) as zero, and the analytical solution can be written as

$$\mathbf{W} = \lambda_2\mathbf{Y}\widetilde{\mathbf{U}}^T\left(\lambda_2\widetilde{\mathbf{U}}\widetilde{\mathbf{U}}^T + \lambda_3\mathbf{I}\right)^{-1} \tag{3.15}$$

(5) Update dictionary $\mathbf{\Omega}$:

Before updating the dictionary, we need to use the inverse matrix reshaping operator $\mathcal{R}^{-1}$ defined in Definition 3 to transform the coefficient $\widetilde{\mathbf{U}}$ to $\overline{\mathbf{U}}$, i.e., $\overline{\mathbf{U}} = \mathcal{R}^{-1}(\widetilde{\mathbf{U}})$. The subproblem for the update of $\mathbf{\Omega}$ is as follows

$$\begin{aligned} &\arg \min_{\mathbf{\Omega}} \frac{1}{2}\|\mathbf{\Omega}\mathbf{X} - \overline{\mathbf{U}}\|_F^2 + \frac{\lambda_4}{2}\|\mathbf{\Omega}\|_F^2 \\ &\text{s.t. } \|\boldsymbol{\omega}_i\|_2^2 \leq 1, \forall i = 1, \ldots, m. \end{aligned} \tag{3.16}$$

The solution to the above problem can be estimated by considering the corresponding unconstrained problem and then enforcing the constraints on atoms of the dictionary by normalizing the atoms. In particular, by setting the gradient to zero, one can get

$$\mathbf{\Omega} = \overline{\mathbf{U}}\mathbf{X}^T \left(\mathbf{X}\mathbf{X}^T + \lambda_4 \mathbf{I}\right)^{-1}. \tag{3.17}$$

To satisfy the constraints $\|\boldsymbol{\omega}_i\|_2^2 \leq 1, \forall i = 1, \ldots, m$, the atoms of $\mathbf{\Omega}$ that do not meet the constraints are further normalized, that is

$$\boldsymbol{w}_i = \frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|_2}, \text{ if } \|\boldsymbol{w}_i\|_2^2 > 1, \ \forall i = 1, \ldots, m. \tag{3.18}$$

The iteration of the algorithm is terminated when either the maximum number of iterations is reached or the change of the loss function between two consecutive iterations is lower than a pre-defined threshold. The proposed algorithm DADL-AOLP is summarized in Algorithm 1. The optimal analysis dictionary $\mathbf{\Omega}^*$ and the classifier $\mathbf{W}^*$ are trained using the algorithm. In the test stage, the coefficient matrix $\overline{\mathbf{U}}^*$ of the testing data $\mathbf{X}_{test}$ can be calculated as $\overline{\mathbf{U}}^* = \mathbf{\Omega}^*\mathbf{X}_{test}$, and then the classification results can be obtained via $\mathbf{Y}^* = \mathbf{W}^*\mathcal{R}(\overline{\mathbf{U}}^*)$ where Y_{ij}^* gives the response of the j-th image to the i-th category. Each testing sample is classified to the category corresponding to the maximum response.

4. Time Complexity

Each iteration of the proposed algorithm consists of the update of five variables, which are the Laplacian matrix $\mathbf{L}$, the coefficient matrices $\overline{\mathbf{U}}$ and $\widetilde{\mathbf{U}}$, the

Algorithm 1 DADL-AOLP

Input: Training set $\mathbf{X}$, regularization parameters λ_1, λ_2, λ_3, λ_4, μ, step size ρ, and maximum number of iterations T.

Output: $\mathbf{\Omega}$ and $\mathbf{W}$

Initialization: The initial $\mathbf{\Omega}$ is set as a random matrix. The iteration index is initialized as $t = 1$.

while not converged **and** $t < T$ **do**

Update Laplacian matrix $\mathbf{L}$ via (2.9), (2.10), (3.5), and (3.6);

Update $\overline{\mathbf{U}}$ via (3.10);

$\widetilde{\mathbf{U}} = \mathcal{R}(\overline{\mathbf{U}})$;

Update $\widetilde{\mathbf{U}}$ via (3.12) and (3.13);

Update $\mathbf{W}$ via (3.15)

$\overline{\mathbf{U}} = \mathcal{R}^{-1}(\widetilde{\mathbf{U}})$

Update $\mathbf{\Omega}$ via (3.17) and (3.18);

$t = t + 1$;

end while

classifier $\mathbf{W}$, and the analysis dictionary $\mathbf{\Omega}$. For convenience, these five steps are denoted as update-$\mathbf{L}$, update-$\overline{\mathbf{U}}$, update-$\widetilde{\mathbf{U}}$, update-$\mathbf{W}$, and update-$\mathbf{\Omega}$, respectively. The update of these variables mainly involves matrix multiplication and matrix inversion. The time complexity of update-$\mathbf{L}$ is $O(m^2s^2)$. Update-$\overline{\mathbf{U}}$ consumes $O\left(m^3\right) + O\left(mnps^2\right) + O\left(m^2np\right)$. Update-$\widetilde{\mathbf{U}}$ consumes $O\left(mnpct\right)$ with t being the iteration number of the inner-loop of this step. Update-$\mathbf{W}$ costs $O\left(m^2p^2n\right) + O\left(m^3p^3\right) + O\left(mnpc\right) + O\left(m^2p^2c\right)$, and update-$\mathbf{\Omega}$ consumes $O\left(mnps^2\right)+O\left(nps^4\right)+O\left(s^6\right)+O\left(ms^4\right)$. Based on this, the total time complexity of the proposed algorithm is $O(m^2s^2)+O\left(mnps^2\right)+O\left(m^2np\right)+O\left(mnpct\right)+O\left(m^2p^2n\right)+O\left(m^3p^3\right)+O\left(m^2p^2c\right)+O\left(nps^4\right)+O\left(s^6\right)+O\left(ms^4\right)$, after omitting higher-order terms.

5. Experimental Results

Experiments are performed to evaluate the proposed algorithm. The databases and settings used in the experiments are presented in Section 5.1, and the convergence of the algorithm is demonstrated in Section 5.2. Experiments in Sections 5.3 and 5.4 aim to investigate the influence of algorithm parameters on the classification performance. Section 5.5 illustrates the classification effect of the proposed algorithm, and Section 5.6 visualizes the effect of the AOLP term. Section 5.7 compares the proposed algorithm with existing related methods to demonstrate the advantages of the proposed algorithm[4].

5.1. Databases and settings

Four widely used classification databases are utilized in the experiments: Extended YaleB [22], UCF-50 [23], Caltech101 [24], and Scene15 [25]. The detailed settings of the databases in the experiments are as follows.

1) Extended YaleB: This database contains face images captured under different illumination conditions. There are 2414 face images of 38 people, and the size of the images is 48×42 pixels. In the experiments, each image is projected to a vector of 504 dimensions via a random matrix. In all experiments, 32 images randomly selected from the samples of each person are used as training data and the remaining as the test data.

2) UCF-50: UCF-50 is a challenging database for action recognition, which contains 50 categories of human actions and each category has $6,680$ videos from YouTube. The videos corresponding to each category are divided into 25 groups, and there are more than 4 action clips in each group. The action bank features [36] are extracted and then reduced to the $5,000$ dimension via principal component analysis (PCA) [37]. In the experiments, the samples of each category are randomly divided into five parts. Four of them are employed as the training set and the remaining part is used as the testing set.

[4]The code of the proposed algorithm is available from `https://github.com/wk-image-code/DADL_AOLP`.

3) Caltech101: Caltech101 consists of $9,144$ images belonging to 101 object classes and a background class, and there are 31 to 800 images in each class. In the experiments, features are extracted via the standard methods of bag-of-words and spatial pyramid matching [25], and then reduced to the dimension of $3,000$ by PCA. For each class, 30 images are used for training, and the remaining samples are utilized as test data.

4) Scene15: Scene15 has $4,485$ images from 15 classes and each class contains at least 200 images. The features are extracted in the same way as for Caltech101. 100 images of each category are randomly chosen as the training set with the remaining images as the test set.

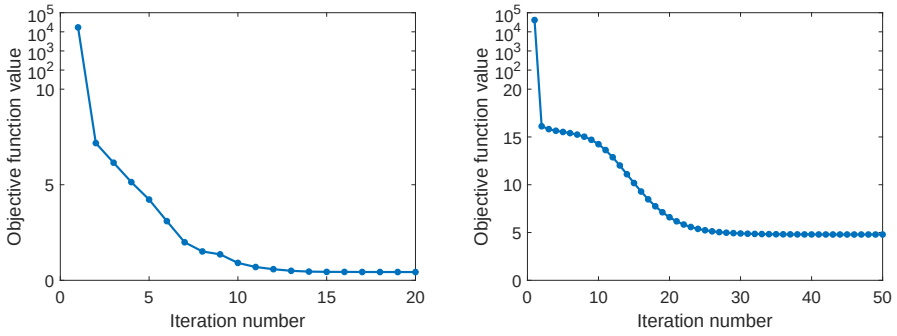
Table 1: Parameter settings of DADL-AOLP for each database.

Parameters	Extended YaleB	UCF-50	Caltech101	Scene15
λ_1	1e-4	1e-4	1e-4	1e-2
λ_2	1e-1	1e-3	1e-2	1e-1
λ_3	1e-1	1e-5	1e-2	1e-4
λ_4	1e-2	1e-2	1e-3	1e-1
μ	1e-4	1	1e-1	1
m	75	50	300	50

The parameter settings of the proposed algorithm including the regularization parameters $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \mu$ and the number of atoms m are given in Table 1. The maximum numbers of iterations are determined empirically according to the change of the objective function values in two consecutive iterations. When the change of the objective function values in two consecutive iterations is smaller than a pre-defined threshold, i.e., $\epsilon = 0.001$, the algorithm is considered as having converged and the current iteration number is set as the maximum iteration number. Based on this, the maximum iterations for the experiments with the datasets Extended YaleB, UCF-50, Caltech101, and Scene15 are set as 45, 20, 50, and 25, respectively.

5.2. Convergence of DADL-AOLP

The UCF-50 and Caltech101 databases are used as examples to illustrate the convergence of DADL-AOLP. The change of the objective function value versus the number of iterations is presented in Fig. 1. For both databases the values of objective function decrease with the increase of the iteration number and converge to stable values after dozens of iterations. It can be seen that, for the UCF-50 dataset, the objective function tends to converge when the iteration number is greater than 15, while for the Caltech101 dataset, the objective function tends to converge when the number of iterations is greater than 35. This demonstrates the convergence of the DADL-AOLP.



(a) Convergence on UCF-50. (b) Convergence on Caltech101.

Figure 1: Convergence curves of DADL-AOLP[5].

5.3. Experiments with different regularization parameters

To evaluate the effect of the regularization parameters on the performance of the proposed method, we take Scene15 as an example and test the algorithm with different settings of the regularization parameters. Specifically, we use different values of λ_1, λ_2 and μ with the other regularization parameters fixed

[5]In the vertical axis of the figure, a logarithmic scale is employed for large values to show the objective function value in the first iteration, and a linear scale is used for the remaining values to show subtle changes of the objective function values in the following iterations.

as $\lambda_3 = 1e-4$, $\lambda_4 = 1e-1$, and the number of atoms m fixed as 50. The results with only one of λ_1, λ_2 and μ fixed as the values in Table 1 are illustrated in Fig. 2 (a)-(c)[6]. From Fig. 2 (a), it can be seen that the proposed algorithm does not perform well when $\lambda_1 = 1$, and when $\lambda_1 \leq 1e-1$, the obtained classification accuracies are improved with the increase of λ_2. Fig. 2 (b) shows that the proposed DADL-AOLP algorithm is relatively insensitive to the settings of λ_1 and μ, and the algorithm achieves the best result when $\mu = 1$. From Fig. 2 (c), we can see that proper settings of λ_2 should be greater than $1e-4$.

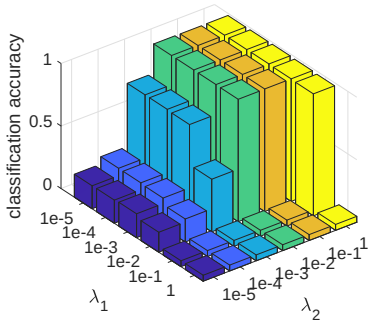
We test the results with different settings of λ_3, λ_4, and μ respectively by fixing the remaining parameters to the values in Table 1, and the results are summarized in Fig. 2 (d)-(f). The results in these sub-figures show that the obtained results are relatively stable when λ_3 and λ_4 are in the range of $1e-6$ and 1. In Fig. 2 (f), we can see that proper settings of μ can improve the classification accuracy effectively, and the proposed algorithm reaches the best performance when $\mu = 1$, which demonstrates the effectiveness of the AOLP term in improving the performance.

5.4. Experiments with different atom numbers

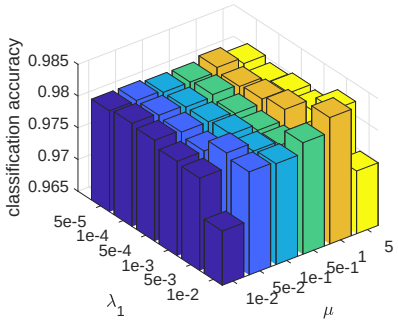
To demonstrate the impact of dictionary size on the results of the proposed algorithm, different numbers of atoms are tested. In particular, the atom numbers are selected from the set {25, 50, 75, 100}. The settings of the other parameters are the same as in Table 1. For comparison, the DCADL algorithm [21] is also tested using different atom numbers, where the other parameters of DCADL are selected according to the original paper. Taking the Scene15 and Extended YaleB databases as examples, the results of DADL-AOLP and DCADL with different atom numbers are shown in Fig. 3.

As can be seen from Fig. 3, in most cases, DADL-AOLP outperforms the DCADL algorithm, which demonstrates the effectiveness of introducing the AOLP

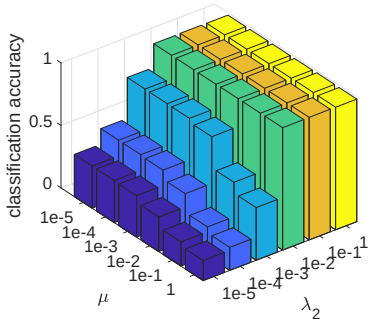
[6]In Fig. 2 (a)-(c), the changing ranges of the varying parameters are determined to show the change of classification accuracies clearly.



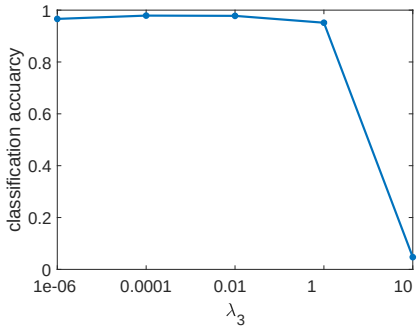
(a) Results with different values of λ_1 and λ_2.



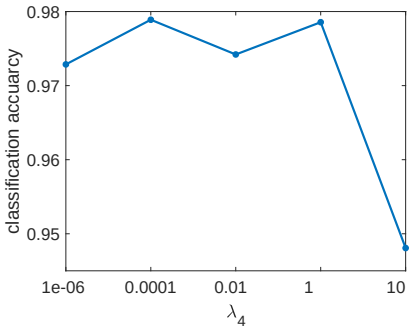
(b) Results with different values of λ_1 and μ.



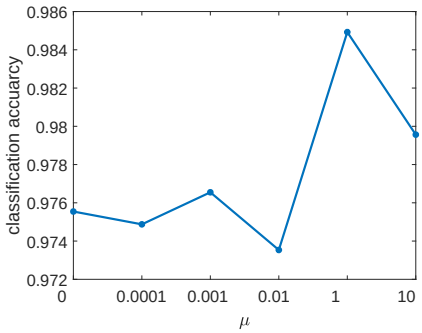
(c) Results with different values of λ_2 and μ.



(d) Results with different values of λ_3.

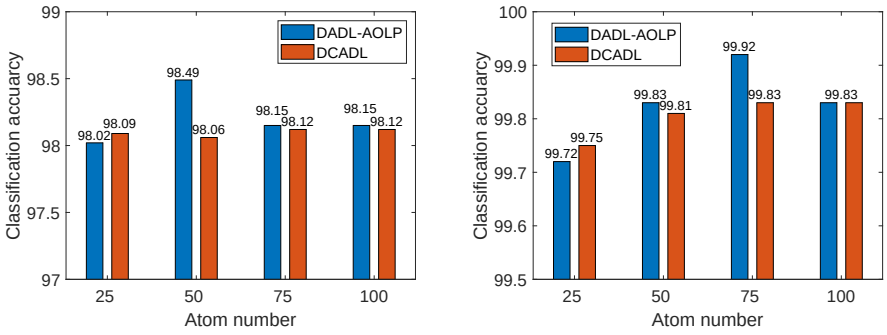


(e) Results with different values of λ_4.



(f) Results with different values of μ.

Figure 2: Classification accuracies obtained by DADL-AOLP with different settings of λ_1, λ_2, λ_3, λ_4 and μ on the Scene15 databse.



(a) Results on Scene15.

(b) Results on Extended YaleB.

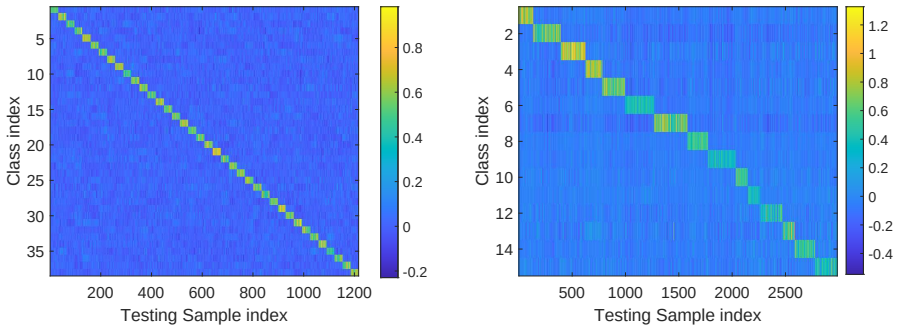
Figure 3: Classification accuracies obtained by DADL-AOLP and DCADL with different atom numbers.

term. As compared with DCADL, the proposed algorithm is more sensitive to the change of atom numbers. This may be because the AOLP term used in DADL-AOLP is based on the distances between atoms and is closely related to the number of atoms.

5.5. Visualization of classification effect

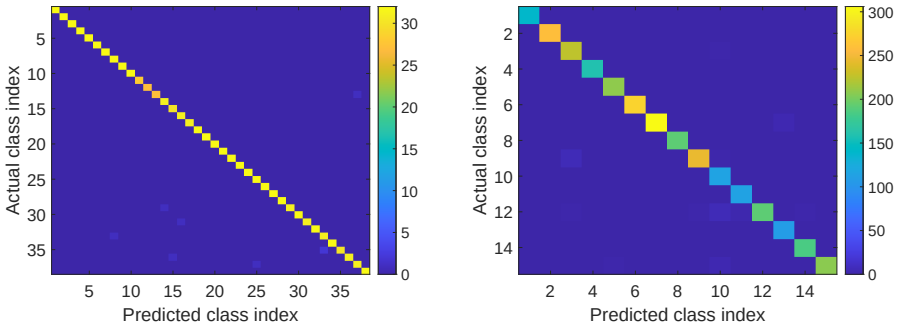
In our algorithm, a linear classifier $\mathbf{W}^*$ and an analysis dictionary $\mathbf{\Omega}^*$ are learned jointly in the training stage, and they are used together to classify testing samples $\mathbf{X}_{test}$ in the testing stage. The analysis dictionary $\mathbf{\Omega}^*$ is utilized to calculate the coding coefficients of $\mathbf{X}_{test}$ and the classifier returns the classification results, i.e. $\mathbf{Y}^* = \|\mathbf{W}^*\mathcal{R}(\mathbf{\Omega}^*\mathbf{X}_{test})\|_F^2$ where Y_{ij}^* denotes the response of the j-th image to the i-th category. Fig. 4 visualizes the elements of $\mathbf{Y}^*$ corresponding to Extended YaleB and Scene 15, where the horizontal axis refers to the indices of testing samples and the vertical axis refers to the indices of classes. One can see that for both databases the classification response matrix $\mathbf{Y}^*$ shows a block diagonal structure, which demonstrates that the proposed algorithm has a good classification effect.

The confusion matrices of the proposed algorithm on Extended YaleB and Scene15 are also visualized in Fig. 5 to illustrate the classification performance.



(a) Extended YaleB. (b) Scene15.

Figure 4: Visualization of classification response matrix $\mathbf{Y}^* = \mathbf{W}^* \mathcal{R}(\mathbf{\Omega}^* \mathbf{X}_{test})$.



(a) Extended YaleB. (b) Scene15.

Figure 5: Visualization of confusion matrices obtained by DADL-AOLP.

In the confusion matrix, the horizontal and vertical coordinates denote the indices of classes in the dataset, and the element in the i-th row and j-th column denotes the number of samples of class i that are classified to class j. It can be seen from Fig. 5 that our proposed algorithm has a good classification effect.

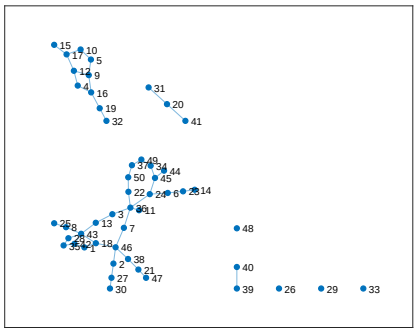
5.6. Effect of the AOLP term

Based on Definition 2, the effect of the AOLP term is to enhance the consistency of dictionary atoms and the corresponding profiles (i.e., rows of the coefficient matrix) in terms of ordinal locality. To show this effect intuitively, the adjacency matrices of the dictionary atoms and the profiles are constructed

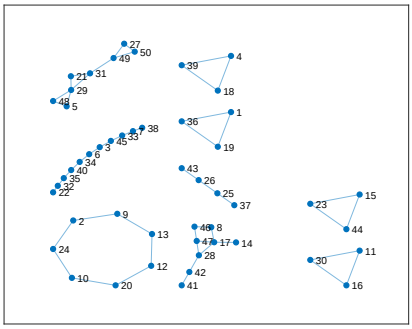
and visualized, respectively. Specifically, the adjacency matrix of the dictionary atoms is constructed as

$$A_{ij} = \begin{cases} 1, & j \in \mathcal{Z}_i, \\ 0, & \text{otherwise}, \end{cases} \tag{5.1}$$

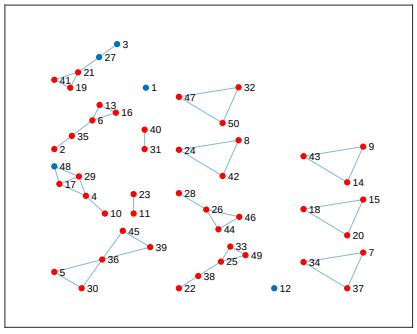
where A_{ij} denotes the (i, j)-th element of the adjacency matrix $\mathbf{A}$, and $\mathcal{Z}_i$ denotes the set of indices of the k nearest atoms of the i-th atom $\boldsymbol{\omega}_i$ based on the Euclidean distance with $k = 2$. The adjacency matrix of the coefficient profiles is also constructed in the same way. The adjacency matrices of the dictionary atoms and the profiles are then visualized using undirected graphs, where the nodes correspond to the atoms or the profiles and the edges connect the nodes with their nearest neighbors.



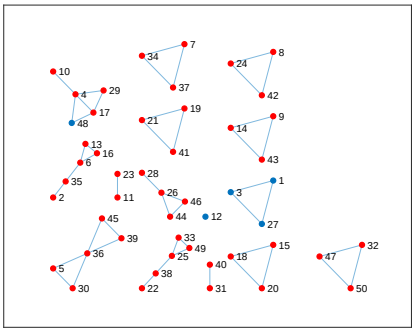
(a) Local adjacency graph of the dictionary atoms by DCADL.



(b) Local adjacency graph of the coefficient profiles by DCADL.



(c) Local adjacency graph of the dictionary atoms by DADL-AOLP.



(d) Local adjacency graph of the coefficient profiles by DADL-AOLP.

Figure 6: Local adjacency graphs by DCADL and DADL-AOLP on UCF-50.

The adjacency graphs of the dictionary atoms and the coefficient profiles by DCADL or the proposed algorithm on UCF-50 are shown in Fig. 6. Note that for clarity, we only visualize the upper triangular matrix of the adjacency matrix. We mark the nodes in the local adjacency graphs with red if they have the same nearest neighbors of the dictionary atoms as those of the coefficient profiles. In the adjacency graphs by the proposed algorithm, i.e., Fig. 6 (c)-(d), the nearest neighbors of most nodes in the local adjacency graphs of the dictionary atoms are the same as those of the coefficient profiles, e.g., nodes $\{7, 34, 37\}, \{15, 18, 20\}$, and $\{9, 14, 43\}$. In contrast, in the adjacency graphs by DCADL, the connections among the nodes vary greatly. This demonstrates that using AOLP in the proposed algorithm can improve the consistency of dictionary atoms and coefficient profiles in terms of the ordinal locality.

5.7. Comparison with other methods

We choose the following algorithms as competing algorithms: DPL [17], SADL [16], RBD-DPL [19], LR-ASDL [20], and DCADL [21]. Among these algorithms, DPL [17] greatly reduces the computational complexity of dictionary learning and can be used to verify the efficiency of the proposed algorithm. SADL [16] is an improved version of the traditional ADL. LR-ASDL [20] and DCADL [21] are the baselines of the proposed method, and RBD-DPL [19] is a recently proposed method.

All experiments adopt the same selection rules for the training and testing data as presented in Section 5.1. The results of DADL-AOLP are obtained based on the parameters shown in Table 1. The classification results of the competing algorithms are from the original papers or obtained based on the released implementation. In particular, for the experiments that have been conducted in the original papers using the same settings as in this paper, we present the results reported in the original papers. For other experiments, we tune the parameters of the competing algorithms and show the best results.

The overall accuracy, recall, precision, and $F1$-score are used to evaluate the classification performance of algorithms. As datasets used in the experiments

contain multiple classes, the weighted macro-average method is employed to calculate the overall results of the precision, recall, and $F1$-score over multiple classes [38], [39]. Given a dataset, let c denote the number of classes, and n_i the number of samples of class i. The number of samples of class i that are classified to class j is denoted as $C(i,j)$. The overall accuracy over all classes and the weighted recall, precision, and $F1$-score for each class i are defined as follows:

$$\begin{aligned}
\text{Accuracy} &= \frac{\sum_{i=1}^{c} C(i,i)}{\sum_{i=1}^{c} n_i}, \\
\text{Recall}(i) &= h_i \frac{C(i,i)}{\sum_{j=1}^{c} C(i,j)}, \\
\text{Precision}(i) &= h_i \frac{C(i,i)}{\sum_{j=1}^{c} C(j,i)}, \\
F1\text{-score }(i) &= \frac{2 \times \text{Precision}(i) \times \text{Recall}(i)}{\text{Precision}(i) + \text{Recall}(i)},
\end{aligned} \tag{5.2}$$

where the weight value h_i represents the ratio of the number of samples of class i to the total number of samples in the dataset, i.e.,

$$h_i = \frac{n_i}{\sum_{i=1}^{c} n_i}. \tag{5.3}$$

The overall recall, precision, and F1-score for a dataset are defined as the sum of the corresponding weighted results of all classes in the dataset. Based on these definitions, it can be seen that the overall recall is equal to the overall accuracy. Therefore, we only present the overall accuracy and omit the overall recall.

5.7.1. Results on Extended YaleB

The recognition rates for Extended YaleB are reported in Table 2. The proposed algorithm obtains the highest classification accuracy among all algorithms. The recognition accuracy obtained by the proposed algorithm is 0.35% higher than that of DCADL and about 2% to 5% higher than the results obtained by the other methods. As compared with DCADL, the proposed algorithm offers performance that is 0.33% higher on precision and 0.36% higher on $F1$-score. The training time of DADL-AOLP is comparable to DCADL but much shorter than LR-ASDL.

Table 2: Classification results (%) on Extended YaleB.

Algorithms	Accuracy	Precision	$F1$-score	Train / Test Time (s)
DPL [17]	97.50	97.71	97.48	7.6 / 5.37e-3
SADL [16]	94.91	96.26	94.30	47.58 / 3.28e-2
RBD-DPL [19]	97.28	97.67	97.18	2.67 / 1.32e-4
LR-ASDL [20]	97.90	98.24	97.83	218.46 / 6.16e-1
DCADL [21]	99.57	99.60	99.56	4.00 / 2.24e-2
DADL-AOLP	**99.92**	**99.93**	**99.92**	10.85 / 2.71e-2

5.7.2. Results on UCF-50

The classification results for the UCF-50 database are summarized in Table 3. As compared with DCADL, the proposed algorithm obtains a 0.3% greater classification accuracy while the training time increases slightly. DADL-AOLP outperforms the remaining competing algorithms significantly in terms of classification performance and is competitive in terms of training and testing time.

Table 3: Classification results (%) on UCF-50.

Algorithms	Accuracy	Precision	$F1$-score	Train / Test Time (s)
DPL [17]	62.90	62.97	60.01	52.67 / 1.72
SADL [16]	70.51	70.91	68.31	431.26 / 1.24
RBD-DPL [19]	67.30	68.32	65.27	47.26 / 0.14
LR-ASDL [20]	68.37	69.87	65.89	624.34 / 3.14
DCADL [21]	76.06	74.85	74.39	56.34 / 0.01
DADL-AOLP	**76.31**	**75.01**	**74.46**	61.14 / 0.01

5.7.3. Results on Caltech101

The classification results for Caltech101 are shown in Table 4. The proposed DADL-AOLP algorithm obtains the best result. As compared with DCADL, the proposed algorithm achieves nearly 1% improvement in accuracy, 0.27% improvement on precision, and 0.69% improvement on $F1$-score. The training and testing time required by the proposed DADL-AOLP algorithm is slightly

longer than that of DCADL and much shorter than that of the other compared algorithms.

Table 4: Classification results (%) on Caltech101.

Algorithms	Accuracy	Precision	$F1$-score	Train / Test Time (s)
DPL [17]	73.90	79.25	73.42	104.33 / 8.13e-1
SADL [16]	72.36	78.37	73.53	936.71 / 8.34e-1
RBD-DPL [19]	72.76	76.84	71.46	67.13 / 9.61e-3
LR-ASDL [20]	65.20	74.31	63.52	871.64 / 8.39e-1
DCADL [21]	74.17	79.87	73.57	32.25 / 6.62e-2
DADL-AOLP	**75.13**	**80.14**	**74.26**	33.60 / 6.87e-2

5.7.4. Results on Scene15

Table 5 shows the classification results on Scene15. As can be seen from the table, LR-ASDL obtains a slightly better result than the proposed algorithm, but its training time is about 55 times of that of DADL-AOLP. The proposed algorithm outperforms the other compared algorithms with relatively shorter training and testing time.

Table 5: Classification results (%) on Scene15.

Algorithms	Accuracy	Precision	$F1$-score	Train / Test Time (s)
DPL [17]	98.10	98.16	98.09	35.22 / 8.10e-2
SADL [16]	98.16	98.23	98.16	202.46 / 1.71e-1
RBD-DPL [19]	98.14	98.20	98.12	17.62 / 1.34e-3
LR-ASDL [20]	**98.60**	**98.63**	**98.59**	381.40 / 5.20e-1
DCADL [21]	98.41	98.49	98.42	5.51 / 1.72e-2
DADL-AOLP	98.49	98.56	98.50	6.92 / 1.44e-2

5.8. Significance test

To compare DCADL and the proposed DADL-AOLP algorithm more completely, we run the algorithms for 10 times using different training sets randomly

selected according to the same allocation ratios as mentioned before. Paired t-test on accuracy, precision, and $F1$-score is then performed, and the results are presented in Table 6.

Table 6: Average results and paired t-test results on accuracy (A), precision (P) and $F1$-score (F) between DADL-AOLP and DCADL.

Datasets		Average Results (%) DADL-AOLP / DCADL	Average Improvement (%)	Standard Deviation	t-value	p-value
Extended YaleB	A	99.72 / 99.57	0.15	0.1464	3.2952	4.70e-3
	P	99.70 / 99.51	0.19	0.0984	6.1278	8.66e-5
	F	99.68 / 99.49	0.19	0.1002	6.1725	8.21e-5
UCF-50	A	75.94 / 75.31	0.63	0.8107	2.4548	1.82e-2
	P	74.99 / 74.44	0.55	0.5344	3.2588	4.90e-3
	F	74.57 / 74.30	0.27	0.4232	2.0412	3.58e-2
Caltech101	A	74.65 / 74.48	0.17	0.1328	3.9919	1.60e-3
	P	80.07 / 79.69	0.38	0.2475	4.8462	4.56e-4
	F	74.00 / 73.38	0.62	0.2051	9.5078	2.72e-5
Scene15	A	98.34 / 97.94	0.40	0.1584	8.0158	1.09e-5
	P	98.40 / 98.22	0.18	0.0765	7.2629	2.37e-5
	F	98.25 / 98.07	0.18	0.0730	7.7101	1.48e-5

Note that the critical t-value for the one-tailed t test with 95% level of confidence and 10 independent tests is 1.8331. The results in Table 6 show that the t-values corresponding to the databases are all larger than 1.8331, and the obtained p-values are all smaller than 0.05. This observation indicates that there is strong evidence that the DADL-AOLP algorithm outperforms DCADL on average and it provides a significant improvement over DCADL.

6. Conclusions and Future Work

A novel discriminative ADL algorithm has been proposed by introducing an AOLP term to the original DCADL model. The AOLP term can enhance the

discrimination capability of coding coefficients by preserving the ordinal locality of dictionary atoms in the learning of dictionary. The proposed algorithm jointly learns a convolutional analysis dictionary together with a linear classifier, and the corresponding optimization has been addressed by updating variables alternatively. The proposed algorithm has been compared with several state-of-the-art dictionary learning methods on four widely used datasets. Experimental results have demonstrated that the AOLP term plays an important role in improving the performance of the discriminative analysis dictionary learning model, and the proposed algorithm performs better than other discriminative dictionary learning algorithms.

One limitation of the proposed algorithm is that it cannot address classification problems in distributed or online settings. This could be an interesting direction for future work. Another limitation is that the parameters are selected empirically, and more sophisticated strategies could be developed in a future work.

7. Acknowledgement

This work was partly supported by the National Natural Science Foundation of China under Grant 61906087 and partly by the Research Center of Security Video and Image Processing Engineering Technology of Guizhou (China) under Grant SRC-Open Project ([2020]001]).

References

[1] B. Li, L. Rencker, J. Dong, Y. Luo, M. D. Plumbley, W. Wang, Sparse analysis model based dictionary learning for signal declipping, IEEE Journal of Selected Topics in Signal Processing 15 (1) (2021) 25–36.

[2] S. Ayas, M. Ekinci, Single image super resolution using dictionary learning and sparse coding with multi-scale and multi-directional gabor feature representation, Information Sciences 512 (2020) 1264–1278.

[3] X. Deng, P. Song, M. R. Rodrigues, P. L. Dragotti, Radar: Robust algorithm for depth image super resolution based on fri theory and multimodal dictionary learning, IEEE Transactions on Circuits and Systems for Video Technology 30 (8) (2019) 2447–2462.

[4] J. Dong, Z. Han, Y. Zhao, W. Wang, A. Prochazka, J. Chambers, Sparse analysis model based multiplicative noise removal with enhanced regularization, Signal Processing 137 (2017) 160–176.

[5] J. Xu, L. Zhang, D. Zhang, A trilateral weighted sparse coding scheme for real-world image denoising, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 20–36.

[6] H. Zheng, H. Yong, L. Zhang, Deep convolutional dictionary learning for image denoising, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 630–641.

[7] Y. Xu, Z. Li, J. Yang, D. Zhang, A survey of dictionary learning algorithms for face recognition, IEEE Access 5 (2017) 8502–8514.

[8] M. Alajmi, K. Awedat, A. Essa, F. Alassery, O. S. Faragallah, Efficient face recognition using regularized adaptive non-local sparse coding, IEEE Access 7 (2019) 10653–10662.

[9] M. Aharon, M. Elad, A. Bruckstein, K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation, IEEE Transactions on Signal Processing 54 (11) (2006) 4311–4322.

[10] J. Dong, W. Wang, W. Dai, M. D. Plumbley, Z.-F. Han, J. Chambers, Analysis simco algorithms for sparse analysis model based dictionary learning, IEEE Transactions on Signal Processing 64 (2) (2016) 417–431.

[11] W. Dai, T. Xu, W. Wang, Simultaneous codeword optimization (SimCO) for dictionary update and learning, IEEE Transactions on Signal Processing 60 (12) (2012) 6340–6353.

[12] I. Ramirez, P. Sprechmann, G. Sapiro, Classification and clustering via dictionary learning with structured incoherence and shared features, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 3501–3508.

[13] Y. Sun, Q. Liu, J. Tang, D. Tao, Learning discriminative dictionary for group sparse representation, IEEE Transactions on Image Processing 23 (9) (2014) 3816–3828.

[14] Z. Jiang, Z. Lin, L. S. Davis, Label consistent K-SVD: Learning a discriminative dictionary for recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (11) (2013) 2651–2664.

[15] S. Hawe, M. Kleinsteuber, K. Diepold, Analysis operator learning and its application to image reconstruction, IEEE Transactions on Image Processing 22 (6) (2013) 2138–2150.

[16] W. Tang, A. Panahi, H. Krim, L. Dai, Structured analysis dictionary learning for image classification, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 2181–2185.

[17] S. Gu, L. Zhang, W. Zuo, X. Feng, Projective dictionary pair learning for pattern classification, Advances in Neural Information Processing Systems 27 (2014) 793–801.

[18] M. Yang, H. Chang, W. Luo, Discriminative analysis-synthesis dictionary learning for image classification, Neurocomputing 219 (2017) 404–411.

[19] Z. Chen, X.-J. Wu, J. Kittler, Relaxed block-diagonal dictionary pair learning with locality constraint for image recognition, IEEE Transactions on Neural Networks and Learning Systems 33 (8) (2022) 3645–3659.

[20] Z. Li, Z. Zhang, J. Qin, S. Li, H. Cai, Low-rank analysis–synthesis dictionary learning with adaptively ordinal locality, Neural Networks 119 (2019) 93–112.

[21] W. Tang, A. Panahi, H. Krim, L. Dai, Analysis dictionary learning: an efficient and discriminative solution, in: 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 3682–3686.

[22] A. S. Georghiades, P. N. Belhumeur, D. J. Kriegman, From few to many: Illumination cone models for face recognition under variable lighting and pose, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 643–660.

[23] K. K. Reddy, M. Shah, Recognizing 50 human action categories of web videos, Machine Vision and Applications 24 (5) (2013) 971–981.

[24] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories, in: 2004 Conference on Computer Vision and Pattern Recognition workshop, IEEE, 2004, pp. 178–178.

[25] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 2, IEEE, 2006, pp. 2169–2178.

[26] M. Li, Y. Guo, M. Li, G. Luo, X. Kong, Coupled dictionary learning for target recognition in SAR images, IEEE Geoscience and Remote Sensing Letters 14 (6) (2017) 791–795.

[27] K. Yan, W. Zheng, Z. Cui, Y. Zong, T. Zhang, C. Tang, Unsupervised facial expression recognition using domain adaptation based dictionary learning approach, Neurocomputing 319 (2018) 84–91.

[28] S. M. Mathews, C. Kambhamettu, K. E. Barner, Maximum correntropy based dictionary learning framework for physical activity recognition using wearable sensors, in: 12th International Symposium Advances in Visual Computing, Springer, 2016, pp. 123–132.

[29] J. Guo, Y. Guo, X. Kong, R. He, Unsupervised feature selection with ordinal locality, in: 2017 IEEE International Conference on Multimedia and Expo (ICME), 2017, pp. 1213–1218.

[30] H. Gao, F. Nie, H. Huang, Local centroids structured non-negative matrix factorization, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, 2017.

[31] M. Chen, X. Li, Concept factorization with local centroids, IEEE Transactions on Neural Networks and Learning Systems 32 (11) (2020) 5247–5253.

[32] M. Chen, X. Li, Robust matrix factorization with spectral embedding, IEEE Transactions on Neural Networks and Learning Systems 32 (12) (2020) 5698–5707.

[33] J. Guo, Y. Guo, X. Kong, M. Zhang, R. He, Discriminative analysis dictionary learning, in: Thirtieth AAAI conference on artificial intelligence, 2016.

[34] Y. Tian, Y. Zhang, A comprehensive survey on regularization strategies in machine learning, Information Fusion 80 (2022) 146–166.

[35] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM Journal on Imaging Sciences 2 (1) (2009) 183–202.

[36] S. Sadanand, J. J. Corso, Action bank: A high-level representation of activity in video, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 1234–1241.

[37] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemometrics and Intelligent Laboratory Systems 2 (1-3) (1987) 37–52.

[38] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Information processing & management 45 (4) (2009) 427–437.

[39] C. Ferri, J. Hernández-Orallo, R. Modroiu, An experimental comparison of performance measures for classification, Pattern recognition letters 30 (1) (2009) 27–38.