Low-resolution image categorization via heterogeneous domain adaptation

Yuan Yao a, Xutao Li a, Yuming Ye a,∗, Feng Liu a, Michael K. Ng b, Zhichao Huang a, Yu Zhang c

a Shenzhen Key Laboratory of Internet Information Collaboration, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China
b Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong
c Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong

ARTICLE INFO
Article history:
Received 20 January 2018
Received in revised form 14 September 2018
Accepted 17 September 2018
Available online xxxx

Keywords:
Low-resolution image categorization
Heterogeneous domain adaptation
Subspace learning

ABSTRACT
Most of existing image categorizations assume that the given datasets have a good resolution and quality. However, the assumption is often violated in real applications. In this paper, we study the low-resolution (LR) image categorization. By utilizing labeled high-resolution (HR) images as auxiliary information, we formulate the problem as a heterogeneous domain adaptation problem and propose a Discriminative Joint Distribution Adaptation (DJDA) model to solve it. The DJDA model projects both LR and HR images into an intermediate subspace with a well-designed objective function, where the distance between classes is expected to be enlarged and the distribution divergence to be reduced. As a result, the discriminative knowledge for HR images can be transferred effectively to LR images. Experimental results demonstrate the proposed DJDA method produces significantly superior categorization accuracies against state-of-the-art competitors.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction
Image categorization, an important task in computer vision and machine learning, has been extensively studied. However, most of existing studies assume that the images are of good resolutions and good qualities. In reality, the assumption is often easily violated. For example, the facial part of a person captured by surveillance cameras tends to be a small low-resolution and noisy image (Fig. 1(left)); similarly, the vehicle images from highway satellites either do not meet a high quality standard (Fig. 1(right)). In security and law-enforcement applications, e.g., criminal identification by surveillance cameras or looking for missing cars by highway satellites, it is extremely useful to classify such low-resolution or low-quality images automatically. However, the problem remains unresolved nicely.

In contrast with conventional image classification, the low-resolution (LR) image categorization poses us two main challenges: (i) due to the limited resolution and the presence of noise, the LR images often lose important discriminative details; and (ii) because of issue (i), the LR training samples are very scarce. One natural answer to the problem is leveraging labeled high-resolution (HR) images for help. As the feature spaces of LR and HR images are heterogeneous, existing work [1–7] exploit them following two lines. On the one hand, [1–3] propose to learn a super-resolution mapping from LR to HR images. By doing so, LR images can be changed into HR ones, and thus classification can be performed in the HR space. However, this kind of methods assume that each LR image in the training set has a HR counterpart, and both of them are obtained under the same imaging conditions, which is obviously too rigorous. On the other hand, LR and HR images are both projected into a common subspace [4–7], and then a classifier is learnt in the subspace. Though the type of methods do not make rigorous assumptions on the same imaging conditions, they still need the correspondence between LR and HR images. In real applications, the correspondence may not be available.

In this paper, we study the LR image categorization problem in a more real-life setting. Our training set comprises of a set of HR images and only a few LR ones for each category. Neither the same imaging conditions nor the direct correspondence between LR and HR images is assumed. We approach the problem in a transfer learning manner [8,9], i.e., borrowing knowledge from HR images for classifying LR images. In particular, the HR images are treated as the source domain and LR images as the target domain. We then aim to adapt the discriminative information in the source domain to the target one. An effective approach called Discriminative Joint Distribution Adaptation (DJDA) is proposed. The main idea of the method is illustrated as in Fig. 2. In DJDA, we first project both the LR and HR images into an intermediate subspace, by enlarging the distance between classes. As a consequence, the discriminative information

https://doi.org/10.1016/j.knosys.2018.09.027
0950-7051/© 2018 Elsevier B.V. All rights reserved.

Please cite this article in press as: Y. Yao, et al., Low-resolution image categorization via heterogeneous domain adaptation, Knowledge-Based Systems (2018).
can be maintained. On the other hand, the distribution difference between LR and HR images is then reduced in the subspace, thereby ensuring an effective knowledge transfer from HR to LR. Finally, LR images in the testbed can be easily categorized by applying nearest neighbor classifier in the intermediate subspace.

The main contributions of the paper are three-fold:

- To the best of our knowledge, we are the first to tackle the LR image categorization problem in a heterogeneous domain adaptation manner, where the correspondence assumption between LR and HR images is removed.
- We propose a simple but effective DJDA approach, which addresses the problem by not only enlarging the distance between classes but also reducing the distribution divergence in the intermediate subspace.
- Extensive experimental results are presented to demonstrate the effectiveness of our DJDA.

The remainder of this paper is organized as follows. We first review related work in Section 2. In Section 3, we then present the proposed model. Experimental settings and results are reported in Section 4. Finally, we conclude the paper in Section 5.

2. Related work

Our work is closely related to two research branches: low-resolution image categorization and heterogeneous domain adaptation.

Low-resolution image categorization. As noted above, existing LR image categorization methods can mainly be grouped into two sub-branches, namely super-resolution (SR) based approaches and projection based approaches. The first sub-branch addresses the classification problem by enhancing the resolution of LR images. For example, Hennings-Yemans et al. [1] proposed Simultaneous Super-Resolution and Recognition ($S^2R^2$) for LR face recognition. Similarly, Zou and Yuen [2] presented a discriminative SR method for LR facial image recognition. In this method, two constraints are designed, namely new data constraint and discriminative constraint, where the former is to ensure the quality of SR images and the latter for maintaining a good separation between classes. Jiang et al. [3] improved the resolution of LR images by developing a Graph Discriminant Analysis on Multi-Manifold (GDAMM) method. However, all the SR methods work by assuming that each LR image has a corresponding HR one, and both of them are taken under the same conditions. The drawback hinders strongly their usability. On the other hand, the second sub-branch projects both LR and HR images into a common subspace for recognition. For instance, in [4], LR and HR images were projected into a common space by minimizing the difference between each LR image and its corresponding HR one. Similarly, Ren et al. [5] leveraged kernel tricks to represent images of different resolutions into a Hilbert space for classification. Biswas et al. [6] utilized Multidimensional Scaling (MDS) to map LR and HR images into a subspace such that the distance between them in the transformed space approximates the distance had both the images been of high resolution. In their subsequent work [7], they further enhanced the method to be robust to different pose conditions. However, for all the projection based methods, each LR image needs a corresponding HR one for training, which is an important drawback. In addition to these approaches, deep learning techniques have been utilized recently in LR image categorization problem, for example, [10,11]. However, the two methods also need the correspondence between LR and HR images. The most closely related method to ours is [12], in which a Joint Multi-scale Discriminant compoEnt Analysis (JUDEA) was proposed for LR person re-identification. Different from the proposed method, JUDEA needs to scale HR (LR) images to LR (HR) ones for learning, which may introduce noises and hurt the accuracy. Our experiments show that the proposed method performs much better than JUDEA.

Heterogeneous Domain Adaptation (HDA). Existing HDA algorithms can be grouped into two categories: symmetric transformation method and asymmetric transformation method. The symmetric transformation method, by learning a pair of projection matrices, aims to transform data of both source and target domains into a common subspace. For example, Wang and Mahadevan [13] presented Domain Adaptation Manifold Alignment (DAMA), which learns projections by maintaining the manifold structure of each domain and discriminative information as well. Duan et al. [14] developed Heterogeneous Feature Augmentation
3. Discriminative joint distribution adaptation

3.1. Problem formulation

We begin with a formal definition of LR image categorization. Let $Y = \{1, 2, \ldots, c\}$ denote the label space, where $c$ is the number of classes. Suppose we are given $n_l$ labeled LR images $\mathcal{L} = \{\mathbf{x}_i^l, y_i\}_{i=1}^{n_l}$ and $n_u$ unlabeled LR images $\mathcal{U} = \{\mathbf{x}_i^u\}_{i=n_l+1}^{n_l+n_u}$ where $\mathbf{x}_i^l \in \mathbb{R}^{m \times 1}$ is the feature of the $i$th LR image, and $y_i \in Y$ is the corresponding class label. Furthermore, assume we have $n_h$ labeled HR images $\mathcal{H} = \{\mathbf{x}_i^h\}_{i=1}^{n_h}$ where $\mathbf{x}_i^h \in \mathbb{R}^{m \times 1}$ denotes the feature of $i$th HR image, and $y_i^h \in Y$ indicates its class label. Note that we have $n_h > m$ and $n_u > n_l$; also, there is no assumption of direct correspondence between LR and HR images. Our goal is to predict the labels of unlabeled LR images in $\mathcal{U}$.

For clarity, we summarize the notations in Table 1.

3.2. Discriminative joint distribution adaptation

As the feature spaces of HR and LR images are different, we cannot directly use HR images to help LR image categorization. Thus, a bridge is needed to bring them together. To this end, we first assume there is an intermediate subspace $\mathbb{R}^{d \times 1}$, and then introduce two projection matrices $\mathbf{P}_1 \in \mathbb{R}^{d \times m}$ and $\mathbf{P}_2 \in \mathbb{R}^{d \times m}$ which are utilized to project any HR $\mathbf{x}^h$ and LR images $\mathbf{x}^l$ into the subspace, respectively.

Our objective is then to find the optimal projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2$. As aforementioned, the projections need take two important factors into account. On the one hand, the distance between images of different classes should be enlarged to ensure discriminative capability. On the other hand, in the intermediate subspace, the divergence between distributions of HR and LR image should be reduced for obtaining an effective transfer. As a result, we propose the following objective function:

$$\min_{\mathbf{P}_1, \mathbf{P}_2} \mathcal{F}(\mathbf{P}_1, \mathbf{P}_2) + \beta \mathcal{D}(\mathbf{P}_1, \mathbf{P}_2) + \lambda \left( \|\mathbf{P}_1\|_F^2 + \|\mathbf{P}_2\|_F^2 \right),$$

where $\mathcal{F}(\cdot, \cdot)$ acts as the discriminative objective, $\mathcal{D}(\cdot, \cdot)$ stands for the distribution alignment, $\beta$ is a positive parameter to balance the importance between the discriminative objective and distribution alignment objective, and $\lambda$ is a positive regularization parameter for preventing over-fitting.

We first introduce how to formulate $\mathcal{F}(\cdot, \cdot)$. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{m \times n}$ be the representation matrix of LR images in $\mathcal{L}$ and $\mathbf{Y} = [y_1, \ldots, y_n] \in \mathbb{R}^{c \times n}$ be the corresponding label matrix, where $\mathbf{f} \in \mathbb{R}^{c \times 1}$ denotes an all-zero vector except for the $i$th component which is one. Similarly, let $\mathbf{X}^h = [\mathbf{x}_1^h, \ldots, \mathbf{x}_n^h] \in \mathbb{R}^{m \times n}$ be the representation matrix of HR images in $\mathcal{H}$ and $\mathbf{Y}^h = [y_1^h, \ldots, y_n^h] \in \mathbb{R}^{c \times n}$ record their label information. Discriminative least square regression [25] has been proven to be an effective tool for multi-class classification. Hence, we formulate $\mathcal{F}(\cdot, \cdot)$ as follows:

$$\mathcal{F}(\mathbf{P}_1, \mathbf{P}_2) = \|\mathbf{X}\mathbf{P}_1 - \mathbf{Y} - \beta \odot \mathbf{S}_l\|_F^2 + \alpha \|\mathbf{X}\mathbf{P}_2 - \mathbf{Y} - \beta \odot \mathbf{S}_u\|_F^2,$$

where $\beta \in \mathbb{R}^{n \times n}$ and $\mathbf{S}_l \in \mathbb{R}^{n \times n}$ are defined as follows:

$$\mathbf{B}_l = \begin{cases} +1, & \text{if } (y_i)_l = 1 \\ -1, & \text{otherwise} \end{cases}, \quad \mathbf{B}_u = \begin{cases} +1, & \text{if } (y_i)_u = 1 \\ -1, & \text{otherwise} \end{cases}.$$

$\mathbf{S}_l \in \mathbb{R}^{n \times n}$ and $\mathbf{S}_u \in \mathbb{R}^{n \times n}$ are nonnegative matrices need to be learned and $\odot$ is a Hadamard product operator of matrices. $\mathcal{F}(\cdot, \cdot)$ comprises of two terms, which are the training losses of LR and HR images, respectively. $\alpha$ is a positive parameter to balance importance of the two parts. The training losses are formulated as...
a least square regression objective, i.e., we aim to regress the label information \( Y_i \) and \( Y_h \) by projecting LR and HR images with matrices \( \mathbf{P}_i \) and \( \mathbf{P}_h \). Importantly, \( \mathbf{B}_i \odot \mathbf{S}_i \) and \( \mathbf{B}_h \odot \mathbf{S}_h \) are the \( \epsilon \)-dropping terms to enlarge the margin between different classes. Note both \( \mathbf{S}_i \) and \( \mathbf{S}_h \) are also unknown variables for estimating. In the objective function, the dimension of our intermediate subspace \( d \) is set to be \( c \).

Next, we introduce how the part \( T(\cdot, \cdot) \) is designed. As the HR and LR images do not have a correspondence, the divergence between their distributions in the intermediate subspace could be extremely large, which is an obstacle for an effective transfer. To avoid the negative situation, we leverage Maximum Mean Discrepancy (MMD) \([26, 16]\) and align the projected distributions. Specifically, the following function is designed:

\[
T(\mathbf{P}_i, \mathbf{P}_h) = \left\| \frac{1}{n_i} \sum_{i=1}^{n_i} \mathbf{x}_i - \frac{1}{n_h} \sum_{j=n_i+1}^{n_i+n_h} \mathbf{x}_j \right\|^2_c + \sum_{j=1}^c \left\| \frac{n_i}{n_i} \mathbf{x}_i - \frac{n_j}{n_j} \mathbf{x}_j \right\|^2_c
\]

\[
= \text{tr}(\mathbf{X}_h \mathbf{x}_h^t) + \sum_{k=1}^c \text{tr}(\mathbf{X}_h \mathbf{x}_h^t)
\]

where \( \mathbf{X}_h = [\mathbf{X}_h^t, \ldots, \mathbf{X}_h^t] \) are the representations of LR and HR images in the intermediate subspace; \( \mathbf{x}_h^t \) denotes the \( j \)th HR (LR) image of class \( k \), \( n_i^k \) is the number of HR (LR) images of class \( k \), and the matrices \( \mathbf{M}_k \in \mathbb{R}^{n_i^k \times n_h^k} \) and \( \mathbf{M}_k \in \mathbb{R}^{n_i^k \times n_h^k} \) are computed as follows:

\[
(M_0)_{ij} = \begin{cases} 
\frac{1}{n_i^k}, & \text{if } i, j \leq n_i^k \\
\frac{1}{n_h^k}, & \text{if } i > n_i^k \\
-\frac{1}{n_h^k}, & \text{ otherwise} 
\end{cases}
\]

\[
(M_k)_{ij} = \begin{cases} 
\frac{1}{n_i^k}, & \text{if } i, j \leq n_i^k \wedge \tilde{j} = \tilde{i} = k \\
\frac{1}{n_h^k}, & \text{if } i > n_i^k \wedge \tilde{j} = \tilde{i} = k \\
\frac{1}{n_h^k}, & \text{if } i < n_i^k, j > n_i^k \wedge \tilde{j} = \tilde{i} = k \\
0, & \text{otherwise} 
\end{cases}
\]

where \( \tilde{i} (\tilde{j}) \) is the label of \( \mathbf{x}_i (\mathbf{x}_j) \). The objective in (4) is composed of alignment losses of joint distributions. On the one hand, the first term adopts the empirical MMD to model the divergence between marginal distributions of projected HR and LR images. Specifically, after projecting HR and LR images into the common subspace, the first term calculates the distance between the centroids of the two parts, and then utilizes this distance to approximate the marginal distribution divergence, which is the exact idea of MMD. On the other hand, the second term applies the modified MMD \([16]\) to approximate the divergence between conditional distributions (i.e., category based distributions). Here the divergence is approximated by calculating the sum of the distance between centroids of the projected HR and LR images in each class. As each centroid is category dependent, the term is conditional distribution related.

Putting all of the above parts together, we have:

\[
\min_{\mathbf{P}_i, \mathbf{P}_h, \mathbf{S}_i, \mathbf{S}_h} \left\| \mathbf{X}_h \mathbf{P}_i - \mathbf{Y}_i - \mathbf{B}_i \odot \mathbf{S}_i \right\|^2_F + \alpha \left\| \mathbf{X}_h \mathbf{P}_h - \mathbf{Y}_h - \mathbf{B}_h \odot \mathbf{S}_h \right\|^2_F + \beta \sum_{k=0}^c \text{tr}(\mathbf{X}_h \mathbf{M}_k \mathbf{M}_k \mathbf{X}_h^t) + \lambda \left( \left\| \mathbf{P}_i \right\|^2_F + \left\| \mathbf{P}_h \right\|^2_F \right)
\]

\[
\text{s.t. } \mathbf{S}_i \geq 0, \mathbf{S}_h \geq 0,
\]

where \( \mathbf{M}_k \in \mathbb{R}^{n_i^k \times n_i^k}, \mathbf{M}_k \in \mathbb{R}^{n_i^k \times n_i^k}, \mathbf{M}_k \in \mathbb{R}^{n_i^k \times n_i^k} \) and \( \mathbf{M}_k \in \mathbb{R}^{n_i^k \times n_i^k} \) are the sub-matrices of \( \mathbf{M}_k \), which are partitioned according to the numbers of HR and LR images.

### 3.3 Optimization

We now detail how to solve the optimization problem (7). An alternating algorithm is applied to optimize the problem, which iteratively updates one variable by fixing the others until convergence. The specific optimization procedure is as follows:

**Fix \( \mathbf{S}_i, \mathbf{S}_h, \mathbf{P}_i, \mathbf{P}_h \) and update \( \mathbf{P}_h \).** In this case, the objective function w.r.t. \( \mathbf{P}_h \), denoted as \( G(\mathbf{P}_h) \), is:

\[
G(\mathbf{P}_h) = \beta \sum_{k=0}^c \text{tr}(\mathbf{X}_h \mathbf{M}_k \mathbf{M}_k \mathbf{X}_h^t) + \alpha \left\| \mathbf{X}_h \mathbf{P}_h - \mathbf{Y}_h - \mathbf{B}_h \odot \mathbf{S}_h \right\|^2_F + \lambda \left\| \mathbf{P}_h \right\|^2_F.
\]

By taking the derivative of (8) w.r.t. \( \mathbf{P}_h \) and setting it to be zero, we can obtain:

\[
\mathbf{P}_h = (\alpha \mathbf{X}_h \mathbf{X}_h^t + \beta \mathbf{M}_k \mathbf{M}_k \mathbf{X}_h^t + \lambda \mathbf{I}_n)^{-1} \left( \alpha \mathbf{X}_h \mathbf{Y}_h + \mathbf{B}_h \odot \mathbf{S}_h \right)
\]

\[
- \beta \mathbf{X}_h \mathbf{M}_k \mathbf{X}_h^t \mathbf{P}_i.
\]

where \( \mathbf{I}_n \in \mathbb{R}^{n \times n} \) is the identity matrix.

**Fix \( \mathbf{S}_i, \mathbf{P}_h, \mathbf{P}_i, \mathbf{S}_h \) and update \( \mathbf{P}_i \):** similarly, the objective function w.r.t. \( \mathbf{P}_i \), denoted as \( G(\mathbf{P}_i) \), is:

\[
G(\mathbf{P}_i) = \beta \sum_{k=0}^c \text{tr}(\mathbf{X}_h \mathbf{M}_k \mathbf{M}_k \mathbf{X}_h^t) + \alpha \left\| \mathbf{X}_h \mathbf{P}_i - \mathbf{Y}_i - \mathbf{B}_i \odot \mathbf{S}_i \right\|^2_F + \lambda \left\| \mathbf{P}_i \right\|^2_F.
\]

By taking the derivative of (10) w.r.t. \( \mathbf{P}_i \) and setting it to be zero, we can obtain:

\[
\mathbf{P}_i = (\alpha \mathbf{X}_h \mathbf{X}_h^t + \beta \mathbf{M}_k \mathbf{M}_k \mathbf{X}_h^t + \lambda \mathbf{I}_n)^{-1} \left( \alpha \mathbf{X}_h \mathbf{Y}_i + \mathbf{B}_i \odot \mathbf{S}_i \right)
\]

\[
- \beta \mathbf{X}_h \mathbf{M}_k \mathbf{X}_h^t \mathbf{P}_h.
\]

where \( \mathbf{I}_n \in \mathbb{R}^{n \times n} \) is the identity matrix.
Algorithm 1 DJDA

Input: Labeled LR images $\mathcal{L} = \{(x_i, y)_i^{\infty}_{i=1}\}$, labeled HR images $\mathcal{H} = \{(x_i, y)_i^{\infty}_{i=1}\}$, and parameters $\alpha, \beta, \lambda$.

Initialize: Randomly initialize $P_0, P_1, S_0, S_1$.
1: Construct $X_0, X_1, Y_0, Y_1, B_0$ and $B_1$ in (7).
2: Calculate $\sum_{k=0}^{n} M_k$ according to (5) and (6).
3: while not converge do
4:   Fix $S_0, S_1, P_0$, update $P_1$ according to (9);
5:   Fix $S_0, S_1, P_1$, update $P_0$ according to (11);
6:   Fix $P_0, P_1$, $S_0$, update $S_1$ according to (13);
7:   Fix $P_0, P_1, S_1$, update $S_0$ according to (15);
8: end while
Output: $P_0$ and $P_1$.

where $D_1 = X_1P_1 - Y_1$.

With the update formulae, our DJDA method can be summarized as in Algorithm 1.

Next, we analyze the time complexity of the DJDA. In line 2, the calculation of $\sum_{k=0}^{n} M_k$ costs $O((n_0 + n_1)^2)$. In each iteration, updating $P_0$ and $P_1$ in lines 4 and 5 takes $O((m_1n_0 + c) + n_1^2 + n_2 + m(n_0 + n_2 + c)^2) + O((m_0n_1 + c) + m_0(n_1 + n_2 + c)^2)$, respectively, the updates of $S_0$ and $S_1$ cost $O((n_0n_1) + C(n_1n_0))$, respectively. As we have $n > m$ and $n_0 \gg n_1 \geq c$, the overall time complexity is $O(n_0^2 + T(n_1^2 + n_0 + n_1^2))$, where $T$ is the number of iterations.

3.4. Prediction

After solving the matrices $P_0$ and $P_1$ by the algorithm, we project both LR and HR images into the intermediate subspace. Categories of LR images in the testbed are predicted by the nearest neighbor classifier in the subspace.

4. Experiments

4.1. Datasets

Due to the lack of large-scale real-world LR benchmark image databases for LR image categorization, we adapt the following four widely-used benchmark image databases to test the effectiveness of our DJDA.

UMIST dataset [27] contains 575 face images of 20 persons. The images of each person were taken under different views, from profile to front. The HR images are obtained by resizing the original pictures into $56 \times 46$ pixels, and LR images are generated by blurring and down-sampling the HR images into $8 \times 7$ pixels. For each person, we use the profile images to generate the LR images and the remaining images to generate HR images.

AR dataset [28] comprises over 4000 color face images collected from 126 people. The pictures were taken with different illumination conditions, facial expressions and disguise conditions. In our experiments, we use a subset of AR, which contains 2600 color face images for 100 people (50 male and 50 female). Similarly, we resize the original images into HR images of $55 \times 40$ pixels, and generate LR images of $8 \times 6$ pixels as UMIST. For each person, we use the images without disguises to generate the HR images and the other images to generate the LR images.

Extended YaleB dataset [29] contains 2414 face images of 38 individuals. The preprocessing is the same as UMIST and AR. Each HR image is represented by $64 \times 56$ pixels, and each LR image contains $8 \times 7$ pixels. For each individual, the images were taken in the negative azimuth which are used to generate LR images and the remaining images to generate the HR images.

COIL-20 dataset [30] consists of 1440 images of 20 objects. Each object was pictured with 72 diverse viewing angles. The HR images and LR images are obtained in the same way as previous three datasets, which contain $64 \times 64$ pixels and $8 \times 8$ pixels, respectively. For each object, we use the images taken in the directions of $0^\circ, 85^\circ, 170^\circ$ to generate the HR images and the other images to generate the LR images.

Table 2 summarizes the statistics of the datasets and Fig. 3 shows some example images. We adopt the pixel values of images as their features. It worth noting that these datasets have a certain degree of diversity and simulate some scenarios in the real world to some extent. Concretely, for UMIST, AR, Extended YaleB and COIL-20 datasets, they simulate the cases that HR and LR images are captured with different views, disguises, illuminations and directions.

4.2. Baselines

We utilize the following eleven baselines for comparison. SVMt: It trains a support vector machine with only the labeled LR images. SVMm2s: It first converts LR images into HR images (called SR images) by utilizing bicubic interpolation, and then trains a support vector machine with both the labeled SR and HR images. SVMs2t: It first directly scales HR images into LR images, and then trains a support vector machine with the labeled images. As mentioned in Section 2, MMDT [19,20], CDSL [22], C-JDA [17] and TNT [23] are state-of-the-art HDA algorithms. We apply them by treating HR images as the source domain and LR images as the target domain. We note that CDSL, G-JDA and TNT are semi-supervised HDA methods, which learn from both labeled and unlabeled data, while DJDA, as a supervised HDA approach, is trained solely upon labeled images. In addition, JUDEA [12] is also compared. In order to study the efficacy of $\mathcal{F}(\cdot, \cdot), \mathcal{F}(\cdot, \cdot)$ and labeled HR images, we evaluate several variants of DJDA: (i) DJDA and DJDa are two reduced DJDA models without $\mathcal{F}(\cdot, \cdot)$ and $\mathcal{F}(\cdot, \cdot)$, respectively; (ii) DJDa learns a reduced DJDA model that $\alpha = 0$ and no projection matrix $P_0$, by only utilizing the labeled LR images.

4.3. Experimental setup and parameter settings

For evaluation, we use the classification accuracy. Given each dataset, we leverage all the labeled HR images and randomly sample $l$ LR images per category to train the models, where $l = 1, 2, 3, 4, 5$ ($l = 3, 6, 9, 12, 15$) for the UMIST and AR (Extended YaleB and COIL-20) datasets. Obviously, the labeled LR images are quite scarce. The rest LR images are used as the testbed. We repeat the sampling process for 10 trials and report the average classification accuracy as final results.
We use LIBLINEAR [31] to run SVM, SVM12s and SVM2t, and implement JUDEA for the comparison. For the other baselines, we experiment with the released codes by their authors. As we have a very limited amount of labeled LR images, cross validation techniques are not suitable for parameter selection. Thus, for a fair comparison, we only tune the parameters of each method on the UMIST dataset with 5 labeled LR images, find the optimal parameter setting and then apply it to other experimental settings and other datasets. The details of the parameter settings of each method are given as follows. For our DJDA, we tune the parameters \((\alpha, \beta, \lambda)\) from \((0.001, 0.01, 0.1, 1, 10, 100, 1000)\), and the best parameter setting is \(\alpha = 0.001, \beta = 0.001\) and \(\lambda = 10\). For DJDA, DJDa and DJDa, we chose their parameters from \((0.001, 0.01, 0.1, 1, 10, 100, 1000)\). For the baselines, we pick the parameters \(C\) in SVM, SVM12s and SVM2t, \((C_s, C_r)\) (see Equation (4) in [19]) in MMDT, \(\lambda\) (see Equation (4) in [22]) in CDLS, \(\lambda\) (see Equation (1) in [17]) in G-JDA, \(\alpha\) (see Equation (5) in [12]) in JUDEA from \((0.001, 0.01, 0.1, 1, 10, 100, 1000)\). For CDLS, the reduced dimensionality is changed from 10 to 40 with an increment of 10, and the ratio \(\delta\) (see Equation (4) in [22]) from 0.1 to 1 with a step size of 0.1. For G-JDA, JUDEA and TNT, the size of common subspace dimensionality varies from 10 to 100 with an increment of 10. For JUDEA, the weight \(\beta\) (see Equation (7) in [12]) is tuned with the values \(0.1, 0.2, \ldots, 1\).

### 4.4. Experimental results

Table 3 reports the performance of all methods on the four datasets. Here \(l = 3\) \((l = 9)\) for UMIST and AR (Extended YaleB and COIL-20) datasets. From this table, we make the following three important observations:

- **DJDA** substantially outperforms all the baseline methods, including naive baselines with SVM, HDA approaches and the variants of DJDA, which demonstrates the superiority and advantages of our DJDA. For SVM, SVM12s and SVM2t, their performances are worse because they either do not use labeled HR images or employ them in a naive manner. Though CDLS and G-JDA utilize the MMD to minimize the distribution divergence as our DJDA, they do not explicitly enlarge the distance between classes in their objective functions. For MMDT and TNT, their inferior performance is because they ignore reducing the distribution divergence during the domain adaptation. Furthermore, though CDLS, G-JDA and TNT utilize both the labeled and unlabeled images, their performance is inferior to our DJDA that is built solely upon labeled ones. As JUDEA directly scales HR (LR) images to LR (HR) ones for learning, which may introduce noises, it performs worse than our DJDA.

- **Contrasting HDA approaches with SVM and DJDa on AR dataset, we find that most of the HDA approaches perform worse than SVM and DJDa.** The observation suggests that adapting knowledge from HR images to LR ones is a hard task on this dataset. However, DJDA yields the best result with an accuracy of \(66.68\%\), which improves the best HDA method G-JDA by \(14.45\%\). This verifies the superiority of DJDA again.

- **We find that the variants, DJDA, DJDa, DJDa, all perform worse than DJDA.** The inferior results of DJDa suggest that the labeled HR images can help LR image categorization and DJDA produces an effective transfer. The worse performance of DJDa and DJDa implies that the two designed mechanisms, reducing distribution divergence \(\mathcal{D}(\cdot, \cdot)\) and enlarging the distance between classes \(\mathcal{F}(\cdot, \cdot)\), are useful. Moreover, comparing the results of DJDA and DJDa, we find that \(\mathcal{F}(\cdot, \cdot)\) is more important than \(\mathcal{D}(\cdot, \cdot)\).

Next, we investigate how the number of labeled LR images per category \(l\) affects the performance. The experimental results are plotted in Fig. 4. We can see from the figure that the proposed DJDA delivers consistently superior performance than all the baseline methods. Furthermore, we find that all the methods tend to perform better as the number of \(l\) is increased, except JUDEA on AR dataset. The reason may be that multi-scale images produced in JUDEA can be unreliable, especially for AR dataset where LR images are disguise ones but HR images are not. As becomes larger, more divergence between LR and HR images will be produced, which hurts the accuracy.

We examine whether the optimal parameters of DJDA are sensitive to experimental settings. As there are three parameters \(\alpha\), \(\beta\) and \(\gamma\), we vary one parameter by keeping the other two fixed, in terms of the default setting \(\{\alpha = 0.001, \beta = 0.001\text{ and }\lambda = 10\}\). Fig. 5, 6, 7 and 8 show the results on UMIST, AR, Extended YaleB and COIL-20, respectively. We can see that, given different number of LR training samples, the best parameter setting is quite consistent. In a word, all the results suggest that the proposed DJDA is quite robust and insensitive to different experiment settings. \(\alpha = 0.001, \beta = 0.001\) and \(\lambda = 10\) can be default parameter values for different applications.

We also conduct experiments to test the convergence of the DJDA. The results are shown in Fig. 9. We can see that the value of our objective function first decreases monotonically and then becomes stable as the number of iterations increases. Also, the accuracy first improves gradually and then barely changes as more iterations are executed. Both suggest the convergence of the proposed method. In terms of the speed, we find that DJDA converges mostly in ten iterations, which is very fast.

Please cite this article in press as: Y Yao et al., Low-resolution image categorization via heterogeneous domain adaptation, Knowledge-Based Systems (2018), https://doi.org/10.1016/j.knosys.2018.09.027.
Fig. 4. Performance of all the methods with different number of labeled LR images per class.

Fig. 5. The parameter tuning test of $\alpha, \beta, \lambda$ on the UMIST dataset. Here $l = 1, 2, 3, 4, 5$ are used.

Fig. 6. The parameter tuning test of $\alpha, \beta, \lambda$ on the AR dataset. Here $l = 1, 2, 3, 4, 5$ are used.

Fig. 7. The parameter tuning test of $\alpha, \beta, \lambda$ on the Extended YaleB dataset. Here $l = 3, 6, 9, 12, 15$ are used.

Finally, we empirically compare the running times of all methods on the UMIST dataset. The results are presented in Fig. 10. We can see that the proposed DJDA is more efficient than all the
Fig. 8. The parameter tuning test of $\alpha$, $\beta$, $\lambda$ on the CoIL-20 dataset. Here $l = 3, 6, 9, 12, 15$ are used.

Fig. 9. Convergence study on four datasets with different number of labeled images.

Fig. 10. The running times of all methods.

domain adaptation approaches (MMMDT, CDLS, G-JDA, JUDEA and TNT). Though DJDA methods are a bit less efficient than conventional methods SVMt and SVMs2t, their accuracies are better as shown above. Compared to SVMt and SVMs2t, SVM2s is very time-consuming, because a bicubic interpolation step is needed.

5. Conclusion

In this paper, we have proposed a simple but effective Discriminative Joint Distribution Adaptation (DJDA) method to address the LR image categorization problem. The DJDA works in a heterogeneous domain adaptation manner, where HR images are treated as the source domain and LR images as the target domain. Two mechanisms are designed in DJDA for an effective knowledge transfer. One is enlarging the distance between classes and the other is reducing the distribution divergence. Extensive experimental results have demonstrated the effectiveness of the proposed DJDA method. In this paper, we utilize the discriminative least square regression, which is a linear method. In the future, we would like to extend the model with kernel functions and to achieve better performance for non-linear cases. In addition, collecting and annotating large-scale real-world LR images to form a benchmark database is also our future interest.

Acknowledgments

This work was supported by the National Key R&D Program of China, 2018YFB0504900, 2018YFB0504905 and the Shenzhen Science and Technology Program, China under Grant JCYJ201708111 60212033, and NSFC, China under Grant Nos. 61602132.

Appendix. Optimization of DJDA

In this section, we detail the specific optimization process of the proposed DJDA. According to (5) and (6), we can see that $M_k$ is symmetric. Thus we have:

$$M_{k1}^T = M_{k1}, M_{k2}^T = M_{k2}, M_{k3}^T = M_{k3}, M_{k4}^T = M_{k4}.$$  \hspace{1cm} (A.1)

The specific optimization procedure is as follows:

**Fix $S_l, S_t, P_l$:** in this case, the objective function w.r.t. $P_h$, denoted as $g(P_h)$, is:

$$g(P_h) = \beta \sum_{k=0}^{c} \text{tr}(P_h^T X_k M_k X_k^T P_h + P_h^T X M_k X^T P_h + P_h^T X_h M_k X_h^T P_l)$$

$$+ \alpha \|X_h^T P_h - Y_h - B_h \circ S_l\|_F^2 + \lambda \|P_h\|_2^2.$$  \hspace{1cm} (A.2)

By taking the derivative of (A.2) w.r.t. $P_h$ and setting it to be 0, we can obtain:

$$\frac{\partial g(P_h)}{\partial P_h} = 0.$$
we use the theorem 2 in [25] to update $S_b$ and yield the following solution:
\[ S_b = \max (B_h \odot D_h, 0). \]  
(A.7)
where $D_h = X_h^T P_h - Y_h$.

Fix $S_b$, $P_h$; the optimization objective w.r.t. $S_z$ is:
\[
\begin{align*}
\min_{S_z} & \quad \|X_z^T P_h - Y_z - B_z \odot S_z\|_F^2, \\
\text{s.t.} & \quad S_z \succeq 0 
\end{align*}
\]  
(A.8)
similar to $S_z$, $S_t$ is derived as:
\[ S_t = \max (B_t \odot D_t, 0). \]  
(A.9)
where $D_t = X_t^T P_t - Y_t$.

References


