A Probabilistic Collaborative Representation based Approach for Pattern Classification

Sijia Cai[1], Lei Zhang[1]*, Wangmeng Zuo[2], Xiangchu Feng[3]
[1]Dept. of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[2]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
[3]Dept. of Applied Mathematics, Xidian University, Xi'an, China
{csscai, cslzhang}@comp.polyu.edu.hk, cswmzuo@gmail.com, xcfeng@mail.xidian.edu.cn

Abstract

Conventional representation based classifiers, ranging from the classical nearest neighbor classifier and nearest subspace classifier to the recently developed sparse representation based classifier (SRC) and collaborative representation based classifier (CRC), are essentially distance based classifiers. Though SRC and CRC have shown interesting classification results, their intrinsic classification mechanism remains unclear. In this paper we propose a probabilistic collaborative representation framework, where the probability that a test sample belongs to the collaborative subspace of all classes can be well defined and computed. Consequently, we present a probabilistic collaborative representation based classifier (ProCRC), which jointly maximizes the likelihood that a test sample belongs to each of the multiple classes. The final classification is performed by checking which class has the maximum likelihood. The proposed ProCRC has a clear probabilistic interpretation, and it shows superior performance to many popular classifiers, including SRC, CRC and SVM. Coupled with the CNN features, it also leads to state-of-the-art classification results on a variety of challenging visual datasets.

1. Introduction

Pattern classification is one of the fundamental problems in computer vision and machine learning. Given a set of training samples $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_K]$, where $\boldsymbol{X}_k, k = 1, 2, \ldots, K$, is the sample matrix of class k, pattern classification aims to predict the class label of a query sample $\boldsymbol{y}$. Many pattern classification schemes have been proposed in the past decades. Generally speaking, there are two categories of pattern classification methods [32, 4]: parametric methods and non-parametric methods. The parametric pattern classification methods (e.g., SVM) focus on how to learn the parameters of a hypothesis classification model from the training data. The learned parametric model is then used to predict the class labels of unknown data. In contrast, the non-parametric pattern classification methods (e.g., nearest neighbor) do not learn a parametric model for classification but use the training samples directly to predict the class labels of unknown data. Though non-parametric methods bear some weaknesses in computational efficiency, recent works have revealed their advantages (e.g., avoid over-fitting) over the parametric based methods [4, 53].

A popular type of non-parametric classifiers which are widely used in various visual recognition tasks are the distance based classifiers, e.g., the nearest subspace classifier (NSC) [10]. The principle of such classifier is to assign a test sample to the class which has the shortest distance to it. However, the distance based non-parametric classifiers rely heavily on the pre-determined distance or similarity metrics. Though some commonly used metrics, such as Euclidean distance, manifold distance and principal angle based correlation [47, 18], are intuitive to describe the variations among samples, they have limitations in accurately reflecting the intrinsic similarity among objects [28]. In order to better characterize the similarity, a promising choice is to introduce the uncertainties of the outputs of a classifier for decision making, as what has been done in probabilistic SVMs [35, 26, 14]. Probabilistic SVM estimates the posterior probabilities of class labels by the calibration techniques, such as Platt's scaling [35, 26] which transforms the classifier's scores into the calibrated probabilities over classes by fitting a sigmoid posterior model.

An alternative approach to probabilistic SVM is the probabilistic subspace methods, e.g., probabilistic principal component analysis (PPCA) [43, 25] and probabilistic linear discriminant analysis (PLDA) [36], which reformulate the subspace methods as a latent variable model and op-

*This work is supported by Hong Kong RGC GRF grant (PolyU 152124/15E).

timize the parameters via maximum likelihood estimation. Therefore, the probabilistic subspace methods can be used to better model the class-conditional densities in classification. Moghaddam and Pentland [30, 31] proposed to utilize a probabilistic similarity measure to model the probability distribution of subspace spanned by the changes of an object's appearance. Wang et al. [47] further extended the probabilistic distance measure from two images to two linear subspaces (image sets), and formulated it as a Bayesian face recognition framework [29]. However, most probabilistic subspace methods make strong assumptions on the distribution of noise and do not provide a straightforward procedure for multi-subspace cases.

How to represent the test sample is a key issue in distance based non-parametric classifiers. In the sparse representation based classifier (SRC) proposed by Wright et al. [48], a test sample is approximated by a linear combination of training samples from all classes with ℓ_1-norm sparsity regularization on the representation coefficients. In [55], Zhang et al. argued that the success of SRC should be largely attributed to the collaborative representation of a test sample by the training samples across all classes. They further proposed an effective collaborative representation based classifier (CRC) by utilizing ℓ_2-norm regularizer. The SRC/CRC classifiers can be regarded as distance based classifiers since they classify a test sample based on the shortest Euclidean distance from it to each class. Many modifications of SRC/CRC have been proposed for face recognition and other visual recognition tasks [49, 46, 12, 8, 9, 21, 54]. Chi and Porikli [8, 9] suggested a collaborative representation optimized classifier (CROC) to combine NSC and SRC/CRC for multi-class classification. Despite the fact that many variants, improvements and applications of SRC/CRC have been proposed, there still lacks a substantial understanding of the classification mechanism of them. Though an inspiring geometric interpretation of CRC has been given in [55], this interpretation is not informative enough to reveal the intrinsic reason of CRC's success.

Motivated by the work of probabilistic subspace methods [30, 31, 28], in this paper we analyze the classification mechanism of CRC from a probabilistic viewpoint and propose a probabilistic collaborative representation based approach for pattern classification. First, we present a probabilistic collaborative representation framework, where the probability that a test sample belongs to the collaborative subspace of all classes can be well defined and computed. Very interestingly, this probabilistic collaborative representation framework explains clearly the ℓ_2-norm regularized representation scheme used in CRC. Consequently, we present a probabilistic collaborative representation based classifier (ProCRC), which jointly maximizes the likelihood that a test sample belongs to each of the multiple classes. The final classification is performed by checking which class has the maximum likelihood. Our extensive experiments on various visual classification tasks demonstrate that ProCRC outperforms many commonly used classifiers, including SVM, kernel SVM, SRC, CRC and CROC.

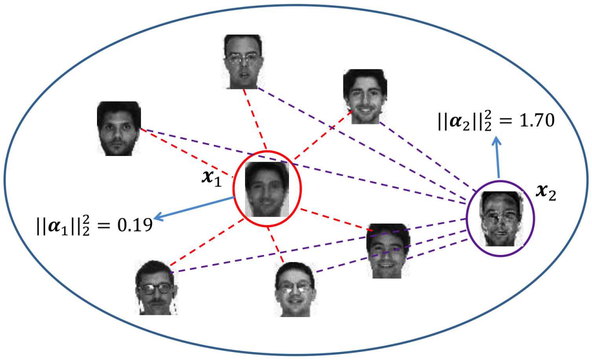
$||\boldsymbol{\alpha}_2||_2^2 = 1.70$
$\boldsymbol{x}_1$
$\boldsymbol{x}_2$
$||\boldsymbol{\alpha}_1||_2^2 = 0.19$

Figure 1. Illustration of probabilistic collaborative subspace. $\boldsymbol{x}_1$ has a smaller ℓ_2-norm of its representation vector, and is more likely to be a face image than $\boldsymbol{x}_2$.

2. Probabilistic Collaborative Subspace Representation

2.1. Probabilistic Collaborative Subspace

Suppose that we have a collection of training samples from K classes $\boldsymbol{X} = [\boldsymbol{X}_1, \cdots, \boldsymbol{X}_K]$, where $\boldsymbol{X}_k$ is the data matrix of class k and each column of $\boldsymbol{X}_k$ is a sample vector. We view $\boldsymbol{X}$ as the data matrix of an expanded class, and denote by $l_{\boldsymbol{X}}$ the label set of all candidate classes in $\boldsymbol{X}$. Denote by $\mathcal{S}$ the linear subspace collaboratively spanned by all samples in $\boldsymbol{X}$. Then for each data point $\boldsymbol{x}$ in the collaborative subspace $\mathcal{S}$, it can be represented as a linear combination of samples in $\boldsymbol{X}$: $\boldsymbol{x} = \boldsymbol{X}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ is the representation vector.

Since $\boldsymbol{X}$ involves many samples from all classes, the collaborative subspace $\mathcal{S}$ is much bigger than the subspace spanned by each individual class $\boldsymbol{X}_k$. Therefore, though all data points $\boldsymbol{X}\boldsymbol{\alpha}$ fall into $\mathcal{S}$, we argue that their confidences to be labeled as $l_{\boldsymbol{X}}$ should be different, depending on how the representation vector $\boldsymbol{\alpha}$ is composed. Let us use an example to explain the idea. As illustrated in Fig. 1, $\boldsymbol{X}$ is a collection of face images from different subjects, and then $l_{\boldsymbol{X}}$ is a label set of face subjects. With vector $\boldsymbol{\alpha}_1 = [0.24,\ 0.22,\ 0.11,\ 0.21,\ 0.13,\ 0.10]$, a face image $\boldsymbol{x}_1 = \boldsymbol{X}\boldsymbol{\alpha}_1$ is composed, and with vector $\boldsymbol{\alpha}_2 = [-0.65,\ 0.46,\ 0.58,\ 0.65,\ -0.42,\ 0.36]$, another face image $\boldsymbol{x}_2 = \boldsymbol{X}\boldsymbol{\alpha}_2$ is composed. Clearly, $\boldsymbol{x}_1$ is more likely to be a face image than $\boldsymbol{x}_2$, and it should have higher confidence to be labeled as $l_{\boldsymbol{X}}$.

From the example in Fig. 1, we can see that the representation vector $\boldsymbol{\alpha}$ determines the confidence that $\boldsymbol{x}$ belongs to $l_{\boldsymbol{X}}$. With a more detailed look of vectors $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$, we can see that $\boldsymbol{\alpha}_1$ contains smaller coefficients (in terms of magnitude), which make $\boldsymbol{x}_1$ approach to the center area of subspace $\mathcal{S}$, while $\boldsymbol{\alpha}_2$ has relatively bigger coefficients, making $\boldsymbol{x}_2$ approach to the boundary area of $\mathcal{S}$. Based on these observations, we propose to formulate $\mathcal{S}$ as a probabilistic collaborative subspace; that is, different data points $\boldsymbol{x}$ have different probabilities of $l(\boldsymbol{x}) \in l_{\boldsymbol{X}}$, where $l(\boldsymbol{x})$ means the label of $\boldsymbol{x}$, and $P(l(\boldsymbol{x}) \in l_{\boldsymbol{X}})$ should be higher if the ℓ_2-norm of $\boldsymbol{\alpha}$ is smaller, vice versa. One intuitive choice is to use a Gaussian function to define such a probability:

$$P\big(l(\boldsymbol{x}) \in l_{\boldsymbol{X}}\big) \propto \exp(-c\|\boldsymbol{\alpha}\|_2^2), \tag{1}$$

where c is a constant. With Eq. (1), we call the subspace $\mathcal{S}$ a probabilistic collaborative subspace, whose data points are assigned different probabilities based on $\boldsymbol{\alpha}$.

2.2. Probabilistic Representation of Samples Outside the Collaborative Subspace

Eq. (1) defines the probability of a data point inside the collaborative subspace $\mathcal{S}$. In practice, the test sample $\boldsymbol{y}$ usually lies outside the subspace $\mathcal{S}$. In order to measure the probability that $\boldsymbol{y}$ belongs to $l_{\boldsymbol{X}}$, i.e., $P(l(\boldsymbol{y}) \in l_{\boldsymbol{X}})$, we could find a data point $\boldsymbol{x}$ in $\mathcal{S}$, and then compute two probabilities: $P(l(\boldsymbol{x}) \in l_{\boldsymbol{X}})$ and the probability that $\boldsymbol{y}$ has the same class label as $\boldsymbol{x}$, i.e., $P(l(\boldsymbol{x}) = l(\boldsymbol{y}))$. With $P(l(\boldsymbol{x}) \in l_{\boldsymbol{X}})$ and $P(l(\boldsymbol{x}) = l(\boldsymbol{y}))$, we can readily have:

$$\begin{aligned} &P\big(l(\boldsymbol{y}) \in l_{\boldsymbol{X}}\big) = \\ &\qquad P\big(l(\boldsymbol{y})=l(\boldsymbol{x})|l(\boldsymbol{x}) \in l_{\boldsymbol{X}}\big) \cdot P\big(l(\boldsymbol{x}) \in l_{\boldsymbol{X}}\big). \end{aligned} \tag{2}$$

$P(l(\boldsymbol{x}) \in l_{\boldsymbol{X}})$ has been defined in Eq. (1). $P\big(l(\boldsymbol{x}) = l(\boldsymbol{y})|l(\boldsymbol{x}) \in l_{\boldsymbol{X}}\big)$ can be measured by the similarity between $\boldsymbol{x}$ and $\boldsymbol{y}$. Here we adopt the Gaussian kernel (a.k.a heat/radial basis function kernel) to define it:

$$P\big(l(\boldsymbol{y}) = l(\boldsymbol{x})|l(\boldsymbol{x}) \in l_{\boldsymbol{X}}\big) \propto \exp(-\kappa\|\boldsymbol{y} - \boldsymbol{x}\|_2^2), \tag{3}$$

where κ is a constant. Gaussian kernel is a widely used measure to characterize the neighbor-based similarity of two vertices in graph, and its advantages have been observed in many real-world applications such as data reduction [17], face analysis [19] and image clustering [56].

With Eq. (1)$\sim$Eq. (3), we have

$$P\big(l(\boldsymbol{y}) \in l_{\boldsymbol{X}}\big) \propto \exp(-(\kappa\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\alpha}\|_2^2 + c\|\boldsymbol{\alpha}\|_2^2)). \tag{4}$$

In order to maximize the probability $P(l(\boldsymbol{y}) \in l_{\boldsymbol{X}})$, we can apply the logarithmic operator to Eq. (4). There is:

$$\begin{aligned} \max P\big(l(\boldsymbol{y}) \in l_{\boldsymbol{X}}\big) &= \max \ln(P\big(l(\boldsymbol{y}) \in l_{\boldsymbol{X}}\big)) \\ &= \min\nolimits_{\boldsymbol{\alpha}} \kappa\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\alpha}\|_2^2 + c\|\boldsymbol{\alpha}\|_2^2 \\ &= \min\nolimits_{\boldsymbol{\alpha}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_2^2 \end{aligned} \tag{5}$$

where $\lambda = c/\kappa$. The above equation gives a probabilistic representation of $\boldsymbol{y}$ over the collaborative subspace $\mathcal{S}$. Interestingly, Eq. (5) shares the same formulation of the representation formula of CRC [55], but it has a clear probabilistic interpretation.

3. The Probabilistic Collaborative Representation based Classifier

Our formulation in Section 2 provides a way to estimate the probability of $l(\boldsymbol{y}) \in l_{\boldsymbol{X}}$ with the collaborative subspace $\mathcal{S}$. However, it cannot indicate which specific class k the sample $\boldsymbol{y}$ belongs to. To perform classification, SRC/CRC simply uses the reconstruction error of $\boldsymbol{y}$ by each class-specific subspace to determine the class label. This classification rule is heuristic and lacks sufficient interpretation. Based on the proposed probabilistic collaborative subspace, in this section we present a probabilistic collaborative representation based classifier (ProCRC) to classify $\boldsymbol{y}$.

3.1. Probability to Each Class-specific Subspace

A sample $\boldsymbol{x} \in \mathcal{S}$ can be collaboratively represented as: $\boldsymbol{x} = \boldsymbol{X}\boldsymbol{\alpha} = \sum_{k=1}^{K} \boldsymbol{X}_k\boldsymbol{\alpha}_k$, where $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1; \boldsymbol{\alpha}_2; \ldots; \boldsymbol{\alpha}_K]$ and $\boldsymbol{\alpha}_k$ is the coding vector associated with $\boldsymbol{X}_k$. Note that $\boldsymbol{x}_k = \boldsymbol{X}_k\boldsymbol{\alpha}_k$ is a data point falling into the subspace of class k. Again by using the Gaussian kernel, the probability that $\boldsymbol{x}$ has the same class label as $\boldsymbol{x}_k$ can be defined as

$$P\big(l(\boldsymbol{x}) = k|l(\boldsymbol{x}) \in l_{\boldsymbol{X}}\big) \propto \exp(-\delta\|\boldsymbol{x} - \boldsymbol{X}_k\boldsymbol{\alpha}_k\|_2^2) \tag{6}$$

where δ is a constant.

For a query sample $\boldsymbol{y}$ outside the space $\mathcal{S}$, we can compute the probability that $l(\boldsymbol{y}) = k$ as:

$$\begin{aligned} P\big(l(\boldsymbol{y})=k\big) &= P\big(l(\boldsymbol{y})=l(\boldsymbol{x})|l(\boldsymbol{x})=k\big) \cdot P(l(\boldsymbol{x})=k) \\ &= P\big(l(\boldsymbol{y})=l(\boldsymbol{x})|l(\boldsymbol{x})=k\big) \cdot \\ &\qquad P\big(l(\boldsymbol{x})=k|l(\boldsymbol{x}) \in l_{\boldsymbol{X}}\big) \cdot P\big(l(\boldsymbol{x}) \in l_{\boldsymbol{X}}\big). \end{aligned} \tag{7}$$

Since the probability definition in Eq. (3) is independent of k as long as $k \in l_{\boldsymbol{X}}$, we have $P\big(l(\boldsymbol{y})=l(\boldsymbol{x})|l(\boldsymbol{x})=k\big) = P\big(l(\boldsymbol{y})=l(\boldsymbol{x})|l(\boldsymbol{x}) \in l_{\boldsymbol{X}}\big)$. With Eq. (5)$\sim$Eq. (7), we have

$$\begin{aligned} P\big(l(\boldsymbol{y})=k\big) &= P\big(l(\boldsymbol{y}) \in l_{\boldsymbol{X}}\big) \cdot P\big(l(\boldsymbol{x})=k|l(\boldsymbol{x}) \in l_{\boldsymbol{X}}\big) \\ &\propto \exp(-(\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_2^2 + \\ &\qquad \gamma\|\boldsymbol{X}\boldsymbol{\alpha} - \boldsymbol{X}_k\boldsymbol{\alpha}_k\|_2^2)), \end{aligned} \tag{8}$$

where $\gamma = \delta/\kappa$.

3.2. The ProCRC Model

By maximizing the probability defined in Eq. (8), we can find some data point $\boldsymbol{x}$ inside $\mathcal{S}$ (or equivalently the representation vector $\boldsymbol{\alpha}$) such that $P(l(\boldsymbol{y}) = k)$ achieves its maximum. However, if we maximize $P(l(\boldsymbol{y}) = k)$ individually for each class k, their corresponding data point $\boldsymbol{x}$ will be different. This makes the classification by the maximal $P(l(\boldsymbol{y}) = k)$ (w.r.t. k) unstable and less discriminative.

Alternatively, a better strategy is that we find a common data point $\boldsymbol{x}$ inside $\mathcal{S}$, which could maximize the joint probability $P(l(\boldsymbol{y}) = 1, \dots, l(\boldsymbol{y}) = K)$. Once the common $\boldsymbol{x}$ is found, we can then check which probability $P(l(\boldsymbol{y}) = k)$ is the highest to determine the class label of $\boldsymbol{y}$. By assuming that the events $l(\boldsymbol{y}) = k$ are independent, we have

$$\begin{aligned}&\max P(l(\boldsymbol{y})=1,\dots,l(\boldsymbol{y})=K)=\max \prod\nolimits_k P(l(\boldsymbol{y})=k)\\ &\propto \max \exp(-(\|\boldsymbol{y} - \boldsymbol{X\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_2^2 +\\ &\qquad \frac{\gamma}{K}\sum\nolimits_{i=1}^{K}(\|\boldsymbol{X\alpha} - \boldsymbol{X}_i\boldsymbol{\alpha}_i\|_2^2))).\end{aligned} \tag{9}$$

Applying the logarithmic operator to Eq. (9) and ignoring the constant term, we have:

$$\begin{aligned}(\hat{\boldsymbol{\alpha}}) = \arg\min\nolimits_{\boldsymbol{\alpha}}\{&\|\boldsymbol{y} - \boldsymbol{X\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_2^2 +\\ &\frac{\gamma}{K}\sum\nolimits_{k=1}^{K}\|\boldsymbol{X\alpha} - \boldsymbol{X}_k\boldsymbol{\alpha}_k\|_2^2\}.\end{aligned} \tag{10}$$

In Eq. (10), the first two terms $\|\boldsymbol{y} - \boldsymbol{X\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_2^2$ form a collaborative representation term, which encourages to find a point $\boldsymbol{x} = \boldsymbol{X\alpha}$ that is close to $\boldsymbol{y}$ in the collaborative subspace $\mathcal{S}$. The last term $\sum_{k=1}^{K}\|\boldsymbol{X\alpha} - \boldsymbol{X}_k\boldsymbol{\alpha}_k\|_2^2$ attempts to find inside each subspace of class k a point $\boldsymbol{X}_k\boldsymbol{\alpha}_k$ which is close to the common point $\boldsymbol{x}$. The parameters γ and λ balance the role of the three terms, which can be set based on the prior knowledge of the problem, or we can use the cross-validation technique to determine γ and λ from the training data. When the regularization parameter $\gamma = 0$, Eq. (10) will degenerate to CRC, and the term $\|\boldsymbol{y} - \boldsymbol{X\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_2^2$ will play a dominant role in determining $\boldsymbol{\alpha}$. When the regularization parameter $\gamma > 0$, the term $\|\boldsymbol{X\alpha} - \boldsymbol{X}_k\boldsymbol{\alpha}_k\|_2^2$ is introduced to further adjust $\boldsymbol{\alpha}_k$ by $\boldsymbol{X_k}$, which results in a more robust and stable solution to $\boldsymbol{\alpha}$.

3.3. The ProCRC Classifier

With the model in Eq. (10), a solution vector $\hat{\boldsymbol{\alpha}}$ is obtained. The probability $P(l(\boldsymbol{y}) = k)$ can be computed by:

$$\begin{aligned}P(l(\boldsymbol{y}) = k) \propto \exp(-(&\|\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\alpha}}\|_2^2 + \lambda\|\hat{\boldsymbol{\alpha}}\|_2^2 +\\ &\frac{\gamma}{K}\|\boldsymbol{X}\hat{\boldsymbol{\alpha}} - \boldsymbol{X}_k\hat{\boldsymbol{\alpha}}_k\|_2^2)).\end{aligned} \tag{11}$$

Note that $(\|\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\alpha}}\|_2^2 + \lambda\|\hat{\boldsymbol{\alpha}}\|_2^2)$ is the same for all classes, and thus we can omit it in computing $P(l(\boldsymbol{y}) = k)$. Let

$$p_k = \exp(-(\|\boldsymbol{X}\hat{\boldsymbol{\alpha}} - \boldsymbol{X}_k\hat{\boldsymbol{\alpha}}_k\|_2^2)). \tag{12}$$

The classification rule can then be formulated as

$$l(\boldsymbol{y}) = \arg\max_k\{p_k\}. \tag{13}$$

We call the above classifier probabilistic collaborative representation based classifier (ProCRC).

3.4. The Robust ProCRC Model

In visual classification, partial corruption or occlusion often degrade the performance. It is well-known that the robustness of classification tasks can be enhanced by using ℓ_1-norm to characterize the loss function [48]. Our proposed probabilistic collaborative representation in Section 2.2 can be easily extended to its robust version. In Eq. (3), we can choose to use the Laplacian kernel, instead of the Gaussian kernel, to measure the probability:

$$P(l(\boldsymbol{y}) = l(\boldsymbol{x}) | l(\boldsymbol{x}) \in l_{\boldsymbol{X}}) \propto \exp(-\kappa\|\boldsymbol{y} - \boldsymbol{x}\|_1). \tag{14}$$

With similar derivations to ProCRC, we can have the following robust ProCRC (R-ProCRC) model:

$$\begin{aligned}(\hat{\boldsymbol{\alpha}}) = \arg\min\nolimits_{\boldsymbol{\alpha}}\{&\|\boldsymbol{y} - \boldsymbol{X\alpha}\|_1 + \lambda\|\boldsymbol{\alpha}\|_2^2 +\\ &\frac{\gamma}{K}\sum\nolimits_{k=1}^{K}\|\boldsymbol{X\alpha} - \boldsymbol{X}_k\boldsymbol{\alpha}_k\|_2^2\}.\end{aligned} \tag{15}$$

The classification rule is the same as that in Eq. (13).

3.5. Solutions to ProCRC and R-ProCRC Models

The proposed ProCRC model has closed form solution, while the proposed R-ProCRC model can be easily solved by the iterative reweighted least square (IRLS) technique.

3.5.1 ProCRC

Refer to Eq. (15), let $\boldsymbol{X}'_k$ be a matrix which has the same size as $\boldsymbol{X}$, while only the samples of $\boldsymbol{X}_k$ will be assigned to $\boldsymbol{X}'_k$ at their corresponding locations in $\boldsymbol{X}$, i.e., $\boldsymbol{X}'_k = [\boldsymbol{0}, \dots, \boldsymbol{X}_k, \dots, \boldsymbol{0}]$. Let $\overline{\boldsymbol{X}}'_k = \boldsymbol{X} - \boldsymbol{X}'_k$. We can then compute the following projection matrix offline:

$$\boldsymbol{T} = (\boldsymbol{X}^T\boldsymbol{X} + \frac{\gamma}{K}\sum\nolimits_{k=1}^{K}(\overline{\boldsymbol{X}}'_k)^T\overline{\boldsymbol{X}}'_k + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T, \tag{16}$$

where $\boldsymbol{I}$ denotes the identity matrix. With $\boldsymbol{T}$, the solution to $\boldsymbol{\alpha}$ can be obtained efficiently:

$$\hat{\boldsymbol{\alpha}} = \boldsymbol{Ty}. \tag{17}$$

3.5.2 R-ProCRC

Though the proposed R-ProCRC model is convex, there is no closed form solution to it, and we adopt an IRLS algorithm to compute $\boldsymbol{\alpha}$.

Based on the current estimation of $\boldsymbol{\alpha}$, we introduce the diagonal weighting matrix $\boldsymbol{W_x}$:

$$\boldsymbol{W_x}(i,i) = 1/|\boldsymbol{X}(i,:)\boldsymbol{\alpha} - \boldsymbol{y}_i|, \tag{18}$$

where $\boldsymbol{X}(i,:)$ refers to the ith row of $\boldsymbol{X}$. Given $\boldsymbol{W_x}$, the problem in Eq. (15) can be reformulated as:

$$\begin{aligned}(\hat{\boldsymbol{\alpha}}) = \arg\min\nolimits_{\boldsymbol{\alpha}}\{&\frac{\gamma}{K}\sum\nolimits_{k=1}^{K}\|\boldsymbol{X\alpha} - \boldsymbol{X}_k\boldsymbol{\alpha}_k\|_2^2 +\\ &\lambda\|\boldsymbol{\alpha}\|_2^2 + (\boldsymbol{X\alpha} - \boldsymbol{y})^T\boldsymbol{W_x}(\boldsymbol{X\alpha} - \boldsymbol{y})\}.\end{aligned} \tag{19}$$

Then the coefficient vector $\boldsymbol{\alpha}$ can be updated by:

$$\hat{\boldsymbol{\alpha}} = (\boldsymbol{X}^T\boldsymbol{W}_{\boldsymbol{x}}\boldsymbol{X}+\frac{\gamma}{K}\sum\nolimits_{k=1}^{K}(\overline{\boldsymbol{X}}'_k)^T\overline{\boldsymbol{X}}'_k+\lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{W}_{\boldsymbol{x}}\boldsymbol{y}. \tag{20}$$

We alternatively update the weighting matrices $\boldsymbol{W}_{\boldsymbol{x}}$ and the coefficient vector $\boldsymbol{\alpha}$, and stop until convergence or after a fixed number of iterations.

4. Experimental results

In this section, we comprehensively evaluate the proposed method from different aspects. In 4.1, by using the (MNIST [20] and USPS [20]) datasets, we compare ProCRC with state-of-the-art representation based classifiers along this line, including NSC [10], SRC [48], CRC [55] and CROC [8, 9]. The linear support vector machine (SVM) classifier [13] is also compared. In 4.2, we compare R-ProCRC with robust SRC [50] on robust face recognition using the AR [27] and Extended Yale B [15] datasets. In 4.3, we evaluate the running time of ProCRC. Finally, in 4.4 we evaluate ProCRC on several challenging visual classification datasets, including Stanford 40 Actions dataset [51], Caltech-UCSD Birds-200-2011 dataset [45], Oxford 102 Flowers dataset [34], Caltech-256 dataset [16] and ImageNet ILSVRC 2012 dataset [38].

The proposed ProCRC has two parameters, λ and γ. In the experiments, we set $\lambda = 10^{-3}$ for handwritten digit datasets and face datasets, and $\lambda = 10^{-2}$ for other datasets. For the parameter γ, we set it by 5-fold cross validation on the training set. For those competing classifiers, their source codes are from the original authors, and we tune their parameters to achieve their best classification accuracy in each experiment.

4.1. Handwritten Digit Recognition

MNIST dataset: The MNIST [20] dataset contains a training set of 60,000 samples and a test set of 10,000 samples. There are 10 classes, and the size of each image is 28×28. We randomly selected 50, 100, 300, and 500 samples from each class for training, and we used all the samples in the test set for testing.

USPS dataset: The USPS [20] dataset also contains a training sample set and a test sample set, and the size of each image is 16×16. We randomly selected 50, 100, 200, and 300 samples from each class for training, and used all the samples in the test set for testing.

Table 1 and Table 2 list the classification rates on the two datasets, respectively. We can see that ProCRC outperforms all the competing classifiers. With the increase of the number of training samples, the classification accuracy of ProCRC increases consistently; however, the classification rate of NSC drops with the increase of training samples, while the rate of CRC first jumps and then increases a little. This shows that ProCRC has good robustness to the number of training samples by considering all the classes collaboratively while double checking each individual class. It has the smallest performance variation under different number of training samples.

Table 1. Classification rate (%) on the MNIST dataset.

Num.	50	100	300	500
SVM	89.35	92.10	94.88	**95.93**
NSC	91.06	92.86	85.29	78.26
CRC	72.21	82.22	86.54	87.46
SRC	80.12	85.63	89.30	92.70
CROC	91.06	92.86	89.93	89.37
ProCRC	**91.84**	**94.00**	**95.48**	95.88

Table 2. Classification rate (%) on the USPS dataset.

Num.	50	100	200	300
SVM	93.46	95.31	95.91	96.30
NSC	93.48	93.25	90.21	87.85
CRC	89.89	91.67	92.36	92.79
SRC	92.58	93.99	95.63	95.86
CROC	93.48	93.25	91.40	91.87
ProCRC	**93.84**	**95.62**	**96.03**	**96.43**

4.2. Face Recognition with Corruption

We then evaluate R-ProCRC for face recognition (FR) with partial occlusion or corruption. The AR [27] and Extended Yale B [15] datasets are used since they are commonly used to in the original papers to evaluate SRC, CRC and CROC. Three types of corruptions are considered: random pixel corruption, random block occlusion, and disguise. In the experiments of random pixel corruption, for each test image we randomly select a certain percentage of pixels and replace them with uniformly distributed values within $[0, 255]$. In the experiments of block occlusion, for each test image we randomly select a square block and replace it with an unrelated image. For real disguise, we use the images with sunglasses or scarf in the AR dataset.

Since the SVM, NSC, CRC and CROC classifiers do not consider the robustness to outliers in design, we only compare R-ProCRC with the robust version (ℓ_1-norm loss function and regularizer) of SRC, denoted by R-SRC [50].

Random corruption: We use the Extended Yale B dataset to evaluate R-ProCRC against random corruption. We randomly selected 30 images from each subject to construct the training dataset, and used the remaining images for testing. Random corruption is added to each test image. Table 3 lists the recognition rates of R-SRC and R-ProCRCr under different ratios of random corruption. One can see that R-ProCRC is much better than R-SRC for FR with random corruption.

Block occlusion: We then compare R-SRC with R-

ProCRC for FR with block occlusion. The same experiment setting as in the random corruption experiment is used by changing random corruption to random corruption. The results are listed in Table 4. One can see that block occlusion will cause more significant performance degradation than random corruption, while R-ProCRC still significantly outperforms R-SRC under different ratios of block occlusion.

Disguise: At last, we use the face images with disguise in the AR dataset to evaluate R-ProCRC. We used the 700 non-occluded images in the first session for training, and used the 600 images with sunglasses and the 600 images with scarf for testing. Table 5 lists the experimental results. Again, R-ProCRC is consistently superior to R-SRC.

Table 3. Recognition rate (%) on face images with random corruption on the Extended Yale B dataset.

Corruption ratio	10%	20%	40%	60%
R-SRC [50]	97.49	95.60	90.19	76.85
R-ProCRC	**98.45**	**98.20**	**93.25**	**82.42**

Table 4. Recognition rate (%) on face images with block occlusion on the Extended Yale B dataset.

Corruption ratio	10%	20%	30%	40%
R-SRC [50]	90.42	85.64	78.89	70.09
R-ProCRC	**98.12**	**92.62**	**86.42**	**77.16**

Table 5. Recognition rate (%) on face images with disguise on the AR dataset.

Corruption ratio	Sunglasses	Scarf
R-SRC [50]	69.17	69.50
R-ProCRC	**70.50**	**69.83**

4.3. Running time comparison

We evaluate the running time of ProCRC and the competing representation based classifiers by processing one test image on the MNIST dataset (5,000 samples for training), and evaluate the running time of R-ProCRC and R-SRC by processing one image on the AR dataset (we test the disguise problem on 600 images with scarf). All methods are implemented in Matlab, and run on a PC with Intel (R) Core (TM) i7-5930K 3.50 GHz CPU and 32 GB RAM. Table 6 lists the running time of different methods.

Since ProCRC and CRC have analytical solutions and the resolved projection matrices have the same size, they have the same speed, which is faster than CROC and much faster than SRC. R-ProCRC employs ℓ_1-norm only for loss function, while R-SRC employs ℓ_1-norm for both loss and regularization. Therefore, R-ProCRC is faster than R-SRC.

Table 6. Running time (s) of different methods.

Method	NSC	CRC	SRC	CROC	ProCRC	R-SRC	R-ProCRC
Time (s)	0.0003	0.0005	0.22	0.0009	0.0005	3.57	1.81

4.4. Other Challenging Visual Classification Tasks

4.4.1 Datasets and settings

To more comprehensively assess the performance of ProCRC, we apply it to four challenging classification datasets: Stanford 40 Actions dataset [51] for action recognition, Caltech-UCSD Birds-200-2011 [45] and Oxford 102 Flowers datasets [34] for fine-grained object recognition, and Caltech-256 dataset [16] for large-scale object recognition. We do not evaluate R-ProCRC since corruption is not the main problem in these datasets.

Stanford 40 Actions dataset [51] is composed of 40 human actions, e.g., brushing teeth, cleaning the floor, reading book, throwing a Frisbee. It contains 9352 images, with 180$\sim$300 images per class. We follow the training-test split settings suggested by the authors [51], using 100 images from each class for training and the remaining for testing.

Caltech-UCSD Birds (CUB200-2011) dataset [45] is a widely-used benchmark for fine-grained image recognition, which contains 11,788 images of 200 bird species. Due to the high degree of similarity among species, this dataset is very challenging. We used the split setting provided in the dataset without part or bounding box annotations. There are around 30 training samples for each species.

Oxford 102 Flowers dataset [34] is another fine-grained image classification benchmark which contains 8,189 images from 102 categories, and each category has at least 40 images. The flowers appear at different scales, pose and lighting conditions. This dataset is challenging since there exist large variations within the category but small difference across several categories.

Caltech-256 dataset [16] consists of 256 object categories with at least 80 images per category. This dataset has a total number of 30,608 images. Following the common experimental settings, we randomly selected 15, 30, 45 and 60 images from each category for training, respectively, and used the remaining images for testing. For fair comparison, we run ProCRC 10 times for each partition and report the average classification accuracy.

On the four datasets, we employ two types of features to demonstrate the effectiveness of ProCRC. First, we use VLFeat [44] to extract the Bag-of-Words feature based on SIFT (refer to BOW-SIFT feature). The square patch size and stride are set as 16×16 and 8 pixels, respectively. The codebook is trained by the k-means method, and the size is 1,024. We use a 2-level spatial pyramid representation.

Table 7. Accuracies (%) of different classifiers with BOW-SIFT features and VGG19 features.

Classifier	Standford 40		CUB200-2011		Flower 102		Caltech 256	
	BOW-SIFT	VGG19	BOW-SIFT	VGG19	BOW-SIFT	VGG19	BOW-SIFT(30)	VGG19(30)
Softmax	21.1	77.2	8.2	72.1	46.5	87.3	25.8	75.3
SVM	24.0	79.0	10.2	75.4	50.1	90.9	28.5	80.1
Kernel SVM	26.3	79.8	**10.5**	76.6	51.0	92.2	28.7	81.3
NSC	22.1	74.7	8.4	74.5	46.7	90.1	25.8	80.2
CRC	24.6	78.2	9.4	76.2	49.9	93.0	27.4	81.1
SRC	24.2	78.7	7.7	76.0	47.2	93.2	26.9	81.3
CROC	24.5	79.1	9.1	76.2	49.4	93.1	27.9	81.7
ProCRC	**28.4**	**80.9**	9.9	**78.3**	**51.2**	**94.8**	**29.6**	**83.3**

The final feature dimension of each image is 5,120 for all datasets. Second, we use VGG-verydeep-19 [42] to extract CNN features (refer to VGG19 features). We use the activations of the penultimate layer as local features, which are extracted from 5 scales $\{2^s, s = -1, -0.5, 0, 0.5, 1\}$. We pool all local features together regardless of scales and locations. The final feature dimension of each image is 4,096 for all datasets. Both BOW-SIFT and VGG19 features are ℓ_2 normalized.

4.4.2 Evaluation of different classifiers with the BOW-SIFT features and CNN feature

To verify that ProCRC is an effective classifier, we present a detailed comparison between ProCRC and several widely-used classifiers, including softmax, linear SVM, kernel SVM with χ^2 kernel, CRC, SRC and CORC. The classification rates on the four datasets with BOW-SIFT features and VGG19 features are listed in Table 7 (the results on Caltech-256 dataset are obtained by using 30 training images per category). From Table 7, we can see that ProCRC almost always achieves the best accuracy with either BOW-SIFT features or VGG19 features among all the classifiers. Specifically, with the powerful CNN features, ProCRC obtains at least 1.5% performance gains over all the other classifiers. These results clearly demonstrate the effectiveness of ProCRC as a visual classifier.

4.4.3 Comparison to state-of-the-art methods

Furthermore, we compare ProCRC (using the VGG19 features) with the state-of-the-art methods on each dataset in Table 8. Note that many of the comparison methods are CNN based methods and their features are even stronger than VGG19.

The classification accuracies on Standford 40 Actions dataset are from SPM [49], LLC [46], EPM [40], Sparse-Bases [51], CF [22], SMP [23] and ASPD [39]. We see that ProCRC achieves at least 5.5% improvement over others. As can be seen in Table 7, using the same VGG19 features, kernel SVM leads to an accuracy of 79.8%, which is 1.1% lower than ProCRC.

The classification accuracies on Caltech-UCSD Birds-200-2011 dataset are from POOF [2], FV-CNN [11], PN-CNN [5] and NAC [41]. Again, ProCRC outperforms all methods except for NAC. However, please note that NAC further constructs a part-model based on the VGG19 feature for recognition, while ProCRC performs classification directly using the VGG19 feature. Compared with the other three methods which all use a specially designed CNN architecture for bird specie recognition, the improvement by ProCRC is obvious.

The classification accuracies on Oxford 102 Flowers dataset are from BiCos seg [6], DAS [1], GMP [33], OverFeat [37] and NAC [41]. ProCRC improves 8% over OverFeat and is only 0.5% lower than NAC, which uses an additional part-model VGG19 feature. The performance gain is significant compared with BiCos seg, DAS and GMP (increase by 15.4%, 14.1% and 10.2%, respectively).

The average classification accuracies (over 10 runs) on Caltech-256 dataset are from ScSPM [49], LLC [46], M-HMP [3], ZF [52], CNN-S [7], VGG19 [42] and NAC [41]. The symbol "-" means that the result is not reported in the original work. ProCRC has at least 12% performance gain over ZF, and has more significant improvements over ScSPM, LLC, M-HMP. When 60 images per class are used for training, ProCRC achieves 1% improvement compared with VGG19 + linear SVM (85.1%), and 2% improvement compared with NAC, while the latter even uses an additional part-model based on VGG19 feature.

4.4.4 The scalability of ProCRC

In the proposed ProCRC model, a matrix inversion operation (see Eq. (16)) will be involved to obtain the projection matrix $\boldsymbol{T}$. The dimensionality of this matrix inverse depends on the number of training samples in the dataset. Therefore, one potential problem of ProCRC is its scalability on very large scale datasets which have millions of training samples (e.g., ImageNet [38]). It might not be fea-

Table 8. Comparsions to state-of-the-arts on different datasets (Standford 40, CUB200-2011, Flower 102 and Caltech-256).

Dataset	Split	Methods & Accuracies (%)							
Standford 40	fixed	ProCRC	ASPD	SMP	CF	SparBases	EPM	LLC	ScSPM
		80.9	75.4	53.0	51.9	45.7	42.2	35.2	34.9
CUB200-2011	fixed	ProCRC	NAC	PN-CNN	FV-CNN	POOF			
		78.3	**81.0**	75.7	66.7	56.9			
Flower 102	fixed	ProCRC	NAC	OverFeat	GMP	DAS	BiCos seg		
		94.8	**95.3**	86.8	84.6	80.7	79.4		
Caltech-256	random	ProCRC	NAC	VGG19	CNN-S	ZF	M-HMP	LLC	ScSPM
	15	**80.2**	-	-	-	65.7	42.7	34.4	27.7
	30	**83.3**	-	-	-	70.6	50.7	41.2	34.0
	45	**84.9**	-	-	-	72.7	54.8	45.3	37.5
	60	**86.1**	84.1	85.1	77.6	74.2	58.0	-	-

sible to load millions of samples into memory and solve a matrix inverse problem with dimensionality of millions.

Fortunately, the scalability problem of ProCRC can be solved by using the dictionary learning (DL) techniques. More specifically, for a dataset which has a large number of samples per class, we can learn a compact dictionary $\boldsymbol{D}_k$, which has only a small number of atoms, from the original samples $\boldsymbol{X}_k$. The ProCRC classifier can then be applied by replacing $\boldsymbol{X}_k$ by $\boldsymbol{D}_k$. One simple DL model is $\min_{\{\boldsymbol{D}_k,\boldsymbol{A}_k\}} \|\boldsymbol{X}_k - \boldsymbol{D}_k\boldsymbol{A}_k\|_F^2 + \tau\|\boldsymbol{A}_k\|_F^2$, where τ is a trade-off parameter and each column of $\boldsymbol{D}_k$ has unit length. This DL model can be easily solved by using an alternating optimization procedure to update $\boldsymbol{D}_k$ and $\boldsymbol{A}_k$.

With the above mentioned DL strategy, we test ProCRC (and other representation based classifiers) on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 dataset [38], which consists of 1.2M+ training images from 1,000 categories (about 1,300 images per category) and 50K validation images (50 images per category). We compare ProCRC with other classifiers using two baseline visual features: BOW-SIFT extracted by VLFeat (we use a codebook of 1,000 visual words to perform the k-means method, and the feature dimension is 1,000 since 0-level spatial pyramid representation is adopted here for simplicity) and AlexNet features extracted by Caffe (as described in [24], the feature dimension is 4,096). For each category, a dictionary with 50 atoms is learned from the about 1,300 samples.

The top-1 and top-5 classification accuracies are listed in Table 9. With the handcraft BOW-SIFT feature, the top-1 accuracy of ProCRC is at least 1.1% higher than all the other competitive classifiers. With the AlexNet based CNN feature, ProCRC outperforms SVM (0.5%) and other representation based classifiers (2%), but is 1.1% and 0.3% lower than Softmax on top-1 and top-5 accuracies, respectively. This is mainly because of the fact that AlexNet features are trained with the Softmax output layer. In summary, DL is effective to solve the scalability issue of ProCRC. In the future, we will explore other methods (e.g., a hierarchical structure) to further improve the performance and scalability of ProCRC.

Table 9. Accuracies (%) on ImageNet ILSVRC-2012.

Classifier	BOW-SIFT		AlexNet	
	top-5	top-1	top-5	top-1
Softmax	28.8	7.4	**80.4**	**57.4**
SVM	29.1	7.2	79.7	55.8
NSC	27.4	6.6	77.4	53.2
CRC	28.3	7.3	78.5	54.3
SRC	28.6	6.9	78.7	54.1
CROC	28.5	7.2	78.8	54.4
ProCRC	**29.7**	**8.5**	80.1	56.3

5. Conclusion

We presented a probabilistic collaborative representation based classifier, namely ProCRC, which employs a probabilistic collaborative representation framework to jointly maximize the probability that a test sample belongs to each class. ProCRC effectively makes use of the training samples from all classes to deduce the class label of a test sample. It possesses a clear probabilistic interpretation, and is very efficient to solve. Our experiments on handwritten digit recognition, face recognition, and other visual classification tasks validated its superiority to popular representation based classifiers, including NSC, CRC, SRC and CROC, as well as benchmark classifiers such as SVM and kernel SVM. Coupled with CNN features (e.g., VGG19), ProCRC demonstrated state-of-the-art performance on challenging visual datasets such as Stanford 40 Actions, CUB200-2011, Oxford 102 Flowers, and Caltech-256. We also demonstrated that ProCRC can be applied to larger-scale dataset such as ImageNet ILSVRC-2012 by introducing a simple dictionary learning pre-processing stage.

References

[1] A. Angelova and S. Zhu. Efficient object detection and segmentation for fine-grained recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 811–818. IEEE, 2013.

[2] T. Berg and P. N. Belhumeur. Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 955–962. IEEE, 2013.

[3] L. Bo, X. Ren, and D. Fox. Multipath sparse coding using hierarchical matching pursuit. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 660–667. IEEE, 2013.

[4] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, pages 1–8. IEEE, 2008.

[5] S. Branson, G. Van Horn, S. Belongie, P. Perona, and C. Tech. Bird species categorization using pose normalized deep convolutional nets. In *BMVC*, 2014.

[6] Y. Chai, V. Lempitsky, and A. Zisserman. Bicos: A bi-level co-segmentation method for image classification. In *International Conference on Computer Vision (ICCV)*, pages 2579–2586. IEEE, 2011.

[7] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014.

[8] Y. Chi and F. Porikli. Connecting the dots in multi-class classification: From nearest subspace to collaborative representation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3602–3609. IEEE, 2012.

[9] Y. Chi and F. Porikli. Classification and boosting with multiple collaborative representations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(8):1519–1531, 2014.

[10] J.-T. Chien and C.-C. Wu. Discriminant waveletfaces and nearest feature classifiers for face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(12):1644–1649, 2002.

[11] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recognition and segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3828–3836, 2015.

[12] W. Deng, J. Hu, and J. Guo. Extended src: Undersampled face recognition via intraclass variant dictionary. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(9):1864–1870, 2012.

[13] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[14] V. Franc, A. Zien, and B. Schölkopf. Support vector machines as probabilistic models. In *Proceedings of the 28th International Conference on Machine Learning*, pages 665–672. ACM, 2011.

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

[16] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007.

[17] J. Ham, D. D. Lee, S. Mika, and B. Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the 21 International Conference on Machine Learning*, page 47. ACM, 2004.

[18] J. Hamm and D. D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 376–383. ACM, 2008.

[19] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang. Face recognition using laplacianfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(3):328–340, 2005.

[20] http://yann.lecun.com/exdb/mnist/. The mnist database of handwritten digits, 2011.

[21] X. Jiang and J. Lai. Sparse and dense hybrid representation via dictionary decomposition for face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(5):1067–1079, 2015.

[22] F. S. Khan, R. M. Anwer, J. van de Weijer, A. D. Bagdanov, A. M. Lopez, and M. Felsberg. Coloring action recognition in still images. *International Journal of Computer Vision*, 105(3):205–221, 2013.

[23] F. S. Khan, J. van de Weijer, R. M. Anwer, M. Felsberg, and C. Gatta. Semantic pyramids for gender and action recognition. *Image Processing, IEEE Transactions on*, 23(8):3633–3645, 2014.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[25] N. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *The Journal of Machine Learning Research*, 6:1783–1816, 2005.

[26] H.-T. Lin, C.-J. Lin, and R. C. Weng. A note on platt's probabilistic outputs for support vector machines. *Machine learning*, 68(3):267–276, 2007.

[27] A. M. Martinez. The ar face database. *CVC Technical Report*, 24, 1998.

[28] B. Moghaddam. Principal manifolds and probabilistic subspaces for visual recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(6):780–788, 2002.

[29] B. Moghaddam, T. Jebara, and A. Pentland. Bayesian face recognition. *Pattern Recognition*, 33(11):1771–1782, 2000.

[30] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *International Conference on Computer Vision (ICCV)*, pages 786–793. IEEE, 1995.

[31] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):696–710, 1997.

[32] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[33] N. Murray and F. Perronnin. Generalized max pooling. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2473–2480. IEEE, 2014.

[34] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, pages 722–729. IEEE, 2008.

[35] J. Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[36] S. J. Prince and J. H. Elder. Probabilistic linear discriminant analysis for inferences about identity. In *International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE, 2007.

[37] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE, 2014.

[38] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[39] F. Shahbaz, J. Xu, J. van de Weijer, A. Bagdanov, M. A. Rao, and A. Lopez. Recognizing actions through action-specific person detection. *Image Processing, IEEE Transactions on*, 24(11):4422–4432, 2015.

[40] G. Sharma, F. Jurie, and C. Schmid. Expanded parts model for human attribute and action recognition in still images. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 652–659. IEEE, 2013.

[41] M. Simon and E. Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *International Conference on Computer Vision (ICCV)*, pages 1143–1151. IEEE, 2015.

[42] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[43] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[44] A. Vedaldi and B. Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the international conference on Multimedia*, pages 1469–1472. ACM, 2010.

[45] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[46] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367. IEEE, 2010.

[47] R. Wang, S. Shan, X. Chen, and W. Gao. Manifold-manifold distance with application to face recognition based on image set. In *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, pages 1–8. IEEE, 2008.

[48] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, 2009.

[49] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pages 1794–1801. IEEE, 2009.

[50] J. Yang and Y. Zhang. Alternating direction algorithms for ℓ_1-problems in compressive sensing. *SIAM journal on scientific computing*, 33(1):250–278, 2011.

[51] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *International Conference on Computer Vision (ICCV)*, pages 1331–1338. IEEE, 2011.

[52] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014.

[53] H. Zhang, A. C. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *Computer Vision and Pattern Recognition (CVPR), 2006 IEEE Conference on*, volume 2, pages 2126–2136. IEEE, 2006.

[54] H. Zhang, J. Yang, Y. Zhang, N. M. Nasrabadi, and T. S. Huang. Close the loop: Joint blind image restoration and recognition with sparse representation prior. In *International Conference on Computer Vision (ICCV)*, pages 770–777. IEEE, 2011.

[55] L. Zhang, M. Yang, and X. Feng. Sparse representation or collaborative representation: Which helps face recognition? In *International Conference on Computer Vision (ICCV)*, pages 471–478. IEEE, 2011.

[56] X. Zheng, D. Cai, X. He, W.-Y. Ma, and X. Lin. Locality preserving clustering for image database. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 885–891. ACM, 2004.