JCBI
Journal of Computing & Biomedical Informatics Volume 05 Issue 02
ISSN: 2710 - 1606 2023
*Research Article*
https://doi.org/10.56979/502/2023

# Dictionary based Bayesian Classifier Learning

**Muhammad Riaz-ud-din[1*], Faisal Shafait[1, 2]**

[1]School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad, Pakistan.
[2]National Center of Artificial Intelligence (NCAI), Islamabad, Pakistan.
[*]Corresponding Author: Muhammad Riaz-ud-din. Email: muhammad.riaz@seecs.edu.pk

**Received:** May 01, 2023 **Accepted:** August 27, 2023 **Published:** September 17, 2023.

**Abstract:** Dictionary and Classifier learning with discriminatory and joint behavior is a considerably effective area in ML research being applied particularly for face recognition, action recognition, and object detection. We present an approach to improve classification performance by enhancing joint learning of the dictionary and classifier. Dictionary and classifier are separately or jointly learned with different sparse representations for training and labels' data. At the perdition stage, sparse representation of a test sample computed over the learned dictionary is used as input for the classifier for classification. The accuracy of the classifier can be increased by using sparse representations of labels over the classifier. To mitigate this issue, we present an approach to jointly learn the same representations for both the test samples and the corresponding labels. At the prediction stage, the computed representation of a test sample over the dictionary will serve the purpose. We performed tests to confirm the effectiveness of our approach, using the Gibbs sampler as an inference for face, object, scene, and action recognition. We compared the results also with other state-of-the-art approaches in the area of Dictionary and Classifier learning. Our approach achieves a classification accuracy significantly higher than that of other approaches.

**Keywords:** Gibbs Sampling; Discriminative; Dictionary and Classifier; Sparse Weights; Sparse Representation; Face Recognition; Object Classification; Action Recognition.

## 1. Introduction

Dictionary learning and sparse representation is remarkably used in different approaches of research areas, particularly in image denoising and restoration [1]–[3], compressive sensing [4]–[6], face recognition [7]–[10], action recognition [11]–[14] etc. A dictionary comprises a few column vectors trained in such a way that examples of a dataset belonging to a particular domain become linear combinations of the column vectors. These column vectors are called basis atoms. For instance, a sample from a specific class of dataset is expressible as a linear combination of the remaining samples of the same class.

A discriminative dictionary may be divided into three types. In the first type, it consists of two types of atoms i.e., the class-specific atoms contributing to the representations of data samples of a particular class, and the atoms contributing to the representations of all samples of the data [15]–[17]. The second type consists of atoms grouped among classes, where each class of atoms represents the data belonging to that class only [18]–[22]. In the third type of dictionaries, a data sample is represented as a linear combination of all the atoms of the dictionary. However, discrimination in the dictionary is induced by making the representations favorable for classification [16], [23].

However, along with inducing discrimination in the dictionary, the relationship between data samples and the corresponding labels is maintained. A Non-parametric Bayesian approach was followed

by [24], [25] to solve the issues in this area in terms of Bayesian settings. They enhanced joint and discriminative behavior while learning the dictionary along with adaptively learning the size of the dictionary. In their approaches, they learned the dictionary and the classifier together and induced joint and discriminative behavior. However, these approaches still have limitations in mapping data samples to the corresponding data labels through the jointly learned classifier. They learn different representations (sparse coefficients) for data samples and the corresponding labels at the dictionary and the classifier learning stages. Whereas, at the prediction stage, representation of a test sample is computed over the dictionary, and this is directly used as input to the classifier for mapping to the corresponding label of the sample. The classifier learned with different representation of the corresponding label cannot perform efficiently by using representation computed over the dictionary for the test sample. To mitigate this issue, we have devised an approach based upon a Non-parametric Bayesian framework that jointly learns the same representations for both the data samples and the corresponding labels at the dictionary and the classifier stages, respectively. Now at the prediction stage, the same representation computed over the dictionary for a test sample will also play the dual role for the representation of the corresponding label over the classifier. Intuitively, this will enhance joint and discriminative learning of the dictionary and will result in increased accuracy of the classification.

We used conjugate priors and analytically derived the posterior probabilities of the proposed Bayesian network. To learn the same representations over the dictionary and the classifier for data samples and the corresponding labels, we draw the representation components from the same base measure.

In our work, we have explained the problem and presented the formulation of our proposed model along with the Gibbs sampler as an inference algorithm to compute posterior probabilities of the latent parameters. Our work also includes the experiments, the comparative results, the discussion, and the conclusion.

**2. Problem Formulation**

A dictionary $\mathbf{\Phi} \in R^{M \times K}$ of $K$ atoms is defined as

$$\mathbf{A} \approx \mathbf{\Phi\alpha} \tag{1}$$

Here, $\mathbf{A} \in R^{M \times N}$ are training examples belonging to $C$ classes i.e., $\mathbf{A}^1, \mathbf{A}^2 \ldots \mathbf{A}^c \ldots \mathbf{A}^C$, indexed in $I_N$. A set $I_c$ contains indices of the training examples belonging to $c^{\text{th}}$ class, or alternately, $\sum_{c=1}^{C} |I_c| = N$, where $|.|$ is the cardinality of a set. $M$ and $N$ represent dimension size and number of examples in the training data. $\boldsymbol{\alpha} \in R^{K \times N}$ denotes the sparse representations of examples in Eq. 1. Similarly, examples belonging to a class can be approximated as

$$\mathbf{A}^c \approx \mathbf{\Phi\alpha}^c \tag{2}$$

Where, $\boldsymbol{\alpha}^c \in R^{K \times |I_c|}$ denotes sparse representations of the examples from $\mathbf{A}^c$. The dictionary and representations coefficients learning is a constrained problem that can be expressed as follows.

$$< \mathbf{\Phi}, \boldsymbol{\alpha} >= \underset{\mathbf{\Phi},\boldsymbol{\alpha}}{min} ||\mathbf{A} - \mathbf{\Phi\alpha}||_F^2 \;\; s.t. \; \forall i, ||\boldsymbol{\alpha}_i||_p \leq t,$$

Here, $\boldsymbol{\alpha}_i \in R^K$ represents the $i^{\text{th}}$ column of $\boldsymbol{\alpha}$ containing representation coefficients of $i^{\text{th}}$ example of training data $\mathbf{A}$. The sparsity of the representation coefficients is controlled by the constant $t$. The $||.||_F$ and $||.||_p$ are Frobenius norm and $l_p$-norm. Following model of a linear classifier can be solved by using the representation coefficients.

$$\mathbf{B} = \underset{\mathbf{B}}{min} \sum_{i=1}^{N} \pounds\{\mathbf{h}_i, f(\boldsymbol{\alpha}_i, \mathbf{B})\} + \lambda ||\mathbf{B}||_F^2,$$

Here, $\mathbf{B} \in R^{C \times K}$, $\lambda$, $f(.)$, £, $\mathbf{h}_i \in \{0,1\}^C$ represent parameters of classifier, regularization constant, predicted label , loss function of the model, and class label for $a_i$ respectively. This model does not integrate classifier with the dictionary learning process and does not exhibit joint learning, as the representation coefficients and the classifier are learned independently. However, joint learning with discriminative behavior of the dictionary and the classifier integrate the labels of the examples in the learning and efficiency of the classifier increases [16], [26], [27]. The approach followed by [24] is very effective among such approaches. Bayesian non-parametric framework with Beta-process was exploited by this approach [28]. [29] introduced the beta process for image restoration and compressive sensing. [25], [24] further made use of this process for object and scene classification, and face and action recognition.

Initially, [28] developed a Beta process for non-parametric factor analysis which is represented by $BP(a_o, b_o, \bar{\mathbf{h}}_o)$ with $a_o > 0, b_o > 0$, and $\bar{\mathbf{h}}_o$ as base measure. Drawing process of Atoms of the dictionary from the base measure can be formulated as.

$$\begin{aligned} \bar{\mathbf{h}} &= \sum_k \pi_k \, \delta_{\boldsymbol{\phi}_k}(\boldsymbol{\phi}),\ k \in K = \{1 \ldots, K\}, \\ \pi_k &\sim \text{Beta}(\pi_k | a_0/K, b_0(K-1)/K), \\ \boldsymbol{\phi} &\sim \bar{\mathbf{h}}_0, \end{aligned}$$

$\delta_{\boldsymbol{\phi}_k}(\boldsymbol{\phi}) = 1$ for $\boldsymbol{\phi} = \boldsymbol{\phi}_k$ and $0$ otherwise. $\bar{\mathbf{h}}$ denotes probabilities expressing frequencies of selection of atoms and its $k^{\text{th}}$ component represents selection frequency of the atom $\boldsymbol{\phi}_k$, drawn from the base measure $\bar{\mathbf{h}}_0$. To induce sparsity, a vector of Bernoulli probabilities,
$\mathbf{B}_r = \{\text{Bernoulli}(\pi_k) : k \in K\}$, corresponding to the atoms selection probabilities $\bar{\mathbf{h}}$ is introduced. Similarly, a sparse binary matrix $\mathbf{Z} \in \{0,1\}^{K \times N}$ of $N$ binary vectors can be drawn for the selection of atoms for all data examples. Due to the sparsity $\mathbf{Z}$, training data can sparsely be approximated as $\mathbf{A} \approx \boldsymbol{\phi}\mathbf{Z}$. Non-zero elements in a column of $\mathbf{Z}$ depend upon the value of $K$. For an instance, if $K \to \infty$, the number of non-zero elements is a draw from Poisson$(\frac{a_0}{b_0})$ distribution [28]. This approach was used by [25] and learned a linear classifier independently. Discrimination in the dictionary was induced by drawing different sets of $\mathbf{B}_r$s for each class of data. However, [24] followed this approach for joint learning of dictionary and classifier. To keep data labels mapped with the data samples, they used the same Bernoulli distributions for both dictionary atoms and classifier atoms selection during joint dictionary and classifier learning. However, they trained different representation coefficients for data samples and the corresponding labels at the dictionary learning and the classifier learning stages respectively. While predicting a test sample, representation coefficients computed over the learned dictionary are used directly as input to the classifier. The different representations learned for the labels during classifier learning get ignored at the prediction stage. We introduced an approach in which this issue is mitigated by learning the same representations at both the stages i.e., at the dictionary learning and the classifier learning stages for both the data examples and the corresponding labels.

**3. Formulation of Our Approach**

We use Non-parametric Bayesian Framework with two Beta-Bernoulli processes. The model is graphically represented in Figure 1 for joint and discriminative learning of the dictionary and classifier. We used different sets of Bernoulli variables with different Bernoulli parameters for each class of the training data. However, the same set of variables for a particular class was used for both the dictionary and the classifier atoms selection for the representation of data examples and the corresponding labels. We formulate our approach as follows. For the construction of the $i^{\text{th}}$ training example of the $c^{\text{th}}$ class it is

formulated as $\mathbf{a}_i^c = \mathbf{\Phi}\boldsymbol{\alpha}_i^c + \mathbf{a}_{\epsilon_i}$ and $\boldsymbol{\alpha}_i^c = \mathbf{z}_i^c \odot \mathbf{s}_i^c$. Here $\boldsymbol{\alpha}_i^c$ is the representation vector for $i^{\text{th}}$ example of data, $\mathbf{s}_i^c \in R^K$ is the weight vector associated with dictionary atoms contributing to sparse code representation of $i^{\text{th}}$ example, and $\odot$ represents the Kronecker product. We learn the same coefficients, $\boldsymbol{\alpha}_i^c$, associated with the classifier $\mathbf{B}$ for the representation of class labels, in contrary to [24]. The formal representation of our model is as below.

$$
\begin{aligned}
&\forall i \in I_c, \forall c \in \{1,2,\dots,C\}, \text{and } \forall k \in \{1,2,\dots,K\} \\
&\boldsymbol{\alpha}_i^c = \mathbf{z}_i^c \odot \mathbf{s}_i^c \\
&\mathbf{a}_i^c = \mathbf{\Phi}\boldsymbol{\alpha}_i^c + \mathbf{a}_{\epsilon_i} \quad \mathbf{h}_i^c = \mathbf{B}\boldsymbol{\alpha}_i^c + \mathbf{h}_{\epsilon_i} \\
&\pi_k^c \sim \text{Beta}(\pi_k^c | a_0/K, b_0(K-1)/K) \\
&z_{ik}^c \sim \text{Bernoulli}(z_{ik}^c | \pi_k^c) \\
&s_{ik}^c \sim \mathcal{N}(s_{ik}^c | 0, 1/\lambda_s^c) \\
&\boldsymbol{\phi}_k \sim \mathcal{N}(\boldsymbol{\phi}_k | \mathbf{0}, 1/\lambda_{\phi_o}\mathbf{I}_M) \quad \mathbf{b}_k \sim \mathcal{N}(\mathbf{b}_k | \mathbf{0}, 1/\lambda_{b_o}\mathbf{I}_C) \\
&\mathbf{a}_{\epsilon_i} \sim \mathcal{N}(\mathbf{a}_{\epsilon_i} | \mathbf{0}, 1/\lambda_a\mathbf{I}_M) \quad \mathbf{h}_{\epsilon_i} \sim \mathcal{N}(\mathbf{h}_{\epsilon_i} | \mathbf{0}, 1/\lambda_h\mathbf{I}_C)
\end{aligned}
\tag{3}
$$

We train the same $\mathbf{z}_i^c$ and $\mathbf{s}_i^c$ for both $\mathbf{a}_i^c$ and $\mathbf{h}_i^c$. We draw $k^{\text{th}}$ coefficients $z_{ik}^c$ and $s_{ik}^c$ of $\mathbf{z}_i^c$ and $\mathbf{s}_i^c$ from Bernoulli distribution and Gaussian distribution respectively. $\lambda$'s are precision parameters of Gaussian distributions as priors in Eq. 3. To represent $k^{\text{th}}$ column of $\mathbf{\Phi}$ and $\mathbf{B}$, we use the notations $\boldsymbol{\phi}_k$ and $\mathbf{b}_k$ respectively. $\mathbf{0}$ is zero vector of dimension $M$ for dictionary prior and of dimension $C$ for classifier prior. The subscript $'o'$ appearing in expressions shows the hyperparameters belonging to prior distributions. We have also modeled errors for the construction of both $\mathbf{a}_i^c$ and $\mathbf{h}_i^c$. We further place non-informative Gamma hyper-priors over precision parameters i.e., $\lambda_s^c \sim \text{Gam}(c_0, d_0)$ and $\lambda_a, \lambda_h \sim \text{Gam}(e_0, f_0)$. The Probabilistic Graphical Model (PGM) of our approach is represented in Figure 1.

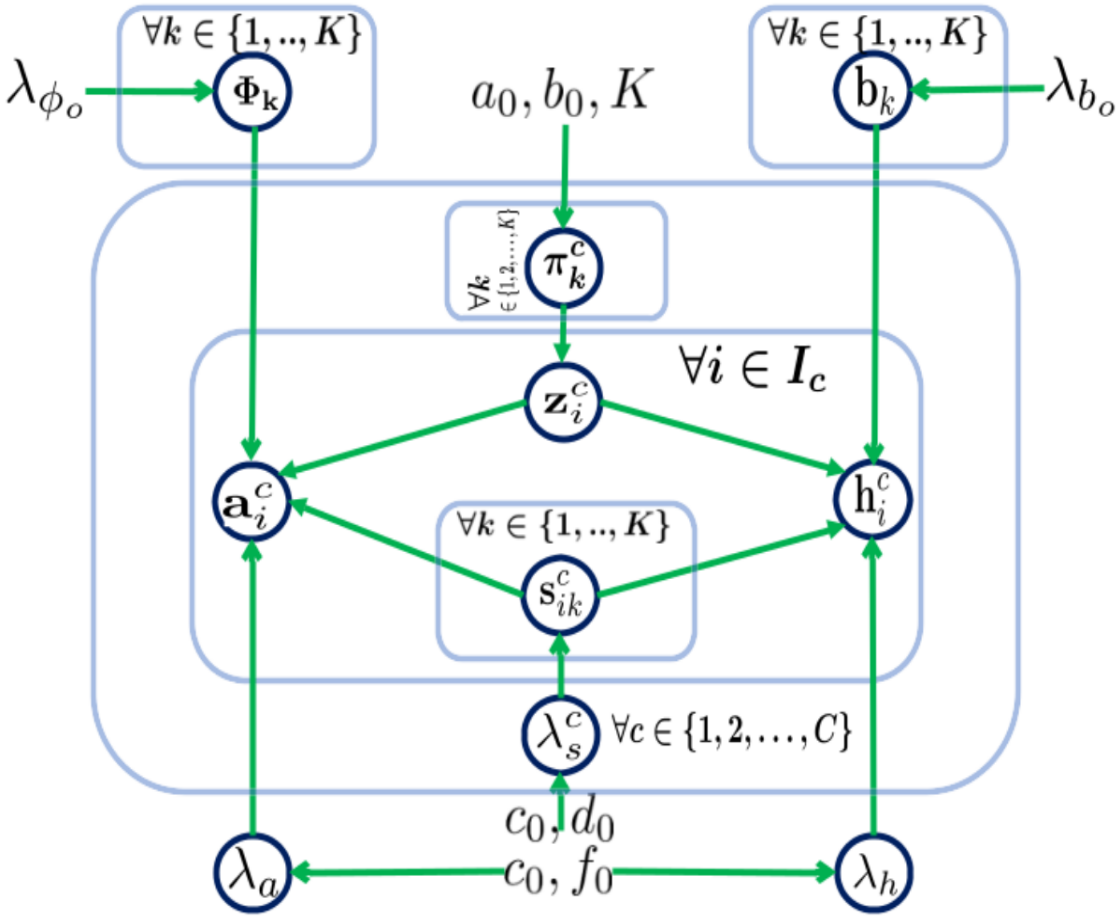


**Figure 1.** Graphical representation of Non-parametric Bayesian Model.

**4. Gibbs Sampling**

Using the Gibbs sampler as an inference algorithm, we iteratively take samples from conditional probabilities for posterior parameters of our model. We drive conditional probabilities for the posterior parameters analytically using conjugate priors in our proposed probabilistic model presented in Eq. 3 and in Figure 1. We have derived the expressions for conditional probabilities for posterior parameters from the joint probability of our model, using the Bayes Theorem. The symbol "| −" in the following conditional probabilities of the posterior variables mean conditioned on all variables except the variable of the mentioned probability.

4.1 Sampling Dictionary Atoms $\boldsymbol{\phi_k}$:

The conditional distribution for taking samples of a dictionary atom may be expressed as

$p(\boldsymbol{\phi}_k|-) \propto \prod_{i=1}^{N} \mathcal{N}(\mathbf{a}_{i_{\phi_k}}|\boldsymbol{\phi}_k(z_{ik}.s_{ik}), \lambda_a^{-1}\mathbf{I}_M)\mathcal{N}(\boldsymbol{\phi}_k|\mathbf{0}, \lambda_{\phi_0}^{-1}\mathbf{I}_M)$ Where,

$\mathbf{a}_{i_{\phi_k}} = \mathbf{a}_i - \boldsymbol{\Phi}(\mathbf{z}_i \odot \mathbf{s}_i) + \boldsymbol{\phi}_k(z_{ik} \odot s_{ik})$, is re-construction error induced by all dictionary atoms except $k^{\text{th}}$ atom in representing $\mathbf{a}_i$. Here dictionary atom does not carry class label $c$ with it, indicating that we are training a dictionary of the third category where all the atoms are shared for the representation of a data example. $\boldsymbol{\phi}_k$ can be sampled from $\mathcal{N}(\boldsymbol{\phi}_k|\boldsymbol{\mu}_k, \lambda_\phi^{-1}\mathbf{I}_M)$, where

$\lambda_\phi = \lambda_{\phi_o} + \lambda_a \sum_{i=1}^{N} (z_{ik}.s_{ik})^2$, $\boldsymbol{\mu}_k = \lambda_a \lambda_\phi^{-1} \sum_{i=i}^{N} (z_{ik}.s_{ik})\mathbf{a}_{i_{\phi_k}}$

4.2 Sampling Classifier Atoms $\mathbf{b_k}$:

Similarly, $\mathbf{b}_k$ can be sampled from $\mathcal{N}(\mathbf{b}_k|\boldsymbol{\mu}_k, \lambda_b^{-1}\mathbf{I}_C)$, where

$\lambda_b = \lambda_{b_o} + \lambda_h \sum_{i=1}^{N} (z_{ik}.s_{ik})^2$, $\boldsymbol{\mu}_k = \lambda_h \lambda_b^{-1} \sum_{i=i}^{N} (z_{ik}.s_{ik})\mathbf{h}_{i_{b_k}}$ Here, $\mathbf{h}_{i_{b_k}}$ is re-construction error induced by all classifier atoms except $k^{\text{th}}$ atom in representing $\mathbf{h}_i$. It may be noted here that we use the same weights for the coefficients of representations, $\mathbf{s}_{ik}$, for both the dictionary and the classifier learning.

4.3 Sampling $\boldsymbol{z_{ik}^c}$ for assignment of atoms:

The conditional probability for the posterior parameter $z_{ik}^c$ can be expressed as

$p(z_{ik}^c|-) \infty \mathcal{N}(\mathbf{a}_{i_{\phi_k}}^c|\boldsymbol{\phi}_k(z_{ik}^c.s_{ik}^c), \lambda_a^{-1}\mathbf{I}_M)$

$\mathcal{N}(\mathbf{h}_{i_{b_k}}|\mathbf{b}_k(z_{ik}^c.s_{ik}^c), \lambda_h^{-1}\mathbf{I}_C)$Bernoulli$(z_{ik}^c|\pi_k^c)$. $z_{ik}^c$ can be sampled from the following:

$z_{ik}^c \sim$ Bernoulli$(\frac{\pi_k^c \zeta_1 \zeta_2}{1-\pi_k^c+\zeta_1\zeta_2\pi_k^c})$, where

$\zeta_1 = \exp(-\frac{\lambda_a}{2}(\boldsymbol{\phi}_k^T\boldsymbol{\phi}_k {s_{ik}^c}^2 - 2s_{ik}^c(\mathbf{a}_{i_{\phi_k}}^c)^T\boldsymbol{\phi}_\mathbf{k}))$ and

$\zeta_2 = \exp(-\frac{\lambda_h}{2}(\mathbf{b}_k^T\mathbf{b}_k {s_{ik}^c}^2 - 2s_{ik}^c(\mathbf{h}_{i_{b_k}}^c)^T\mathbf{b}_\mathbf{k}))$

4.4 Sampling Representation Coefficient Weights $\boldsymbol{s_{ik}^c}$:

The conditional distribution for $s_{ik}^c$ is

$p(s_{ik}^c|-) \propto \mathcal{N}(\mathbf{a}_{i_{\phi_k}}^c|\boldsymbol{\phi}_\mathbf{k}(z_{ik}^c.s_{ik}^c), \lambda_a^{-1}\mathbf{I}_M)$

$\mathcal{N}(\mathbf{h}_{i_{b_k}}^c|\mathbf{b}_k(z_{ik}^c.s_{ik}^c), \lambda_h^{-1}\mathbf{I}_C)N(s_{ik}^c|0,1/\lambda_s^c)$,

The conjugacy relationship makes it possible to derive distribution analytically as given below

$s_{ik}^c \sim \mathcal{N}(s_{ik}^c|\mu_s, \lambda^{-1})$, where:

$\lambda = \lambda_s^c + \lambda_a {z_{ik}^c}^2 \boldsymbol{\phi}_k^T\boldsymbol{\phi}_k + \lambda_h {z_{ik}^c}^2 \mathbf{b}_k^T\mathbf{b}_k$,

$\mu_s = \lambda^{-1}\left(\lambda_a z_{ik}^c \boldsymbol{\phi}_k^T \mathbf{a}_{i_{\phi_k}}^c + \lambda_h z_{ik}^c \mathbf{b}_k^T \mathbf{h}_{i_{b_k}}^c\right)$,

Here the weights for coefficients, $s_{ik}^c$, are learned jointly for the representation of both the training examples and the training labels. This behavior of our approach makes it distinct from others.

4.5 Sampling atoms selection probabilities and pruning atoms $\boldsymbol{\pi_k^c}$:

$p(\pi_k^c|-)$

$\propto \prod_{i\in I_c} \text{Bernoulli}\,(z_{ik}^c|\pi_k^c)\text{Beta}\left(\pi_k^c|\frac{a_o}{K},\frac{b_o(K-1)}{K}\right)$,

$\propto \text{Beta}\left(\frac{a_o}{K}+\sum_{i=1}^{|I_c|} z_{ik}^c, \frac{b_o(K-1)}{K}+|I_c|-\sum_{i=1}^{|I_c|} z_{ik}^c\right)$. A dictionary atom $\phi_k$ is pruned at each iteration of Gibbs sampling according to whether $\sum_{c=1}^{C} \boldsymbol{\pi}_k^c \to 0$ or not [24]. Likewise, classifier atom $\mathbf{b}_k$ is also pruned.

4.6 Sampling of Precision parameters for coefficients $\boldsymbol{\lambda_s^c}$:

$p(\lambda_s^c|-) \propto \prod_{i\in I_c} \mathcal{N}\,(\mathbf{s}_i^c|\mathbf{0}, 1/\lambda_s^c\mathbf{I}_K)\text{Gam}(\lambda_s^c|c_o,d_o)$.

$\lambda_s^c \sim \text{Gam}\left(\frac{|I_c|K}{2}+c_o, \frac{1}{2}\sum_{i=1}^{|I_c|} \parallel s_i^c \parallel_2^2 + d_o\right)$.

4.7 Sampling of Precision Parameter for Data $\boldsymbol{\lambda_a}$:

$p(\lambda_a|-) \propto \prod_{i=1}^{N} \mathcal{N}\,(\mathbf{a}_i|\boldsymbol{\Phi}(\mathbf{z}_i \odot \mathbf{s}_i), \lambda_a^{-1}\mathbf{I}_M)\text{Gam}(\lambda_a|e_o,f_o)$

$\lambda_a \sim \text{Gam}\left(\frac{MN}{2}+e_o, \frac{1}{2}\sum_{i=1}^{N} \parallel \mathbf{a}_i - \boldsymbol{\Phi}(\mathbf{z}_i \odot \mathbf{s}_i) \parallel_2^2 + f_o\right)$

4.8 Sampling of Precision Parameter for Labels $\boldsymbol{\lambda_h}$:

Similarly, $\lambda_h$

$\sim \text{Gam}(\frac{CN}{2}+e_o, \frac{1}{2}\sum_{i=1}^{N} \parallel \mathbf{h_i} - \mathbf{B}(\mathbf{z_i} \odot \mathbf{s_i}) \parallel_2^2 + f_o)$.

After large number of iterations of the Gibbs sampler, we compute the posterior probability distributions of dictionary atoms and the classifier parameters. Sparse representation $\boldsymbol{\alpha}$ for the prediction of a test sample is computed over $\boldsymbol{\Phi}$ first. The label of the test sample is predicted by classifying $\boldsymbol{\alpha}$ with the classifier $\mathbf{B}$. A predicted label corresponding to the test sample is estimated as $\mathbf{B}\boldsymbol{\alpha} \in R^C$ and the index of the largest value is declared as the class label. It may be noted here that the same sparse representation $\boldsymbol{\alpha}$ computed over the dictionary is also the representation of the corresponding label over the classifier. Orthogonal Matching Pursuit (OMP) [30] is used to compute $\boldsymbol{\alpha}$. As the same $\boldsymbol{\alpha}$ is jointly learned at both the dictionary and the classifier stages, we expect that $\mathbf{B}\boldsymbol{\alpha} \in R^C$ will result in the true label, enhancing the class prediction efficiency.

**5. Parameters Initialization:**

The overcomplete dictionary is initialized by randomly selecting a sufficiently large number of samples from the training data, in the order of 1.25 times the data. We use OMP to compute sparse representation coefficients for initializing $\mathbf{s}_i^c$. We initialize $\mathbf{z}_i^c$ with all its components equal to one except those having zero values for their corresponding components of $\mathbf{s}_i^c$, in which case they are set equal to zero. Ridge regression technique is used to initialize the classifier $\mathbf{B}$, using $\mathbf{s}_i^c$ and training labels $\mathbf{h}_i^c \in R^C$ [16], [25], [31]. We set all $\pi_k$ values equal to 0.5 to make the selection of dictionary and classifier atoms equally probable for representation of data samples and the corresponding labels.

**Algorithm 1** Gibbs sampling

**Require:** Initialize the hyperparameters $a_o$, $b_o$ with $0 < a_0, b_0 < \min_c |I_c|$, $c_o$, $d_o$, $e_o$, $f_o$ with $10^{-6}$, $\lambda_{\phi_o}$, $\lambda_{b_o}$ with
$M$ and $C$, $\lambda_s^c$ with 1, and $\lambda_a$ and $\lambda_h$ with $10^9$.
Initialize $\mathbf{\Phi}$, $\mathbf{B}$, $\boldsymbol{\pi}_k^c$, $\mathbf{z}_k^c$, and $\mathbf{s}_k^c$ as explained in the parameters initialization section.

**for** $i \in \{1,2,3,\ldots,500\}$ **do**
  **for** $k \in \{1,2,3,\ldots,K\}$ **do**
    Sample $\boldsymbol{\phi}_k$, $\mathbf{b}_k$, $\mathbf{s}_k^c$, $\mathbf{z}_k^c$, and $\pi_k^c$ (from expressions of conditional distributions ) $\forall c \in \{1,2,3,\ldots,C\}$
    $k = k + 1$
  **end for**
  Sample $\lambda_a$, $\lambda_h$, and $\lambda_s^c$ $(\forall c \in \{1,2,3,\ldots,C\})$
  $i = i + 1$
**end for**
Compute sparse representation coefficients $\boldsymbol{\alpha}$ of test data over the learned dictionary $\mathbf{\Phi}$, using orthogonal matching pursuit (OMP), a module in SPAMS package of Python.
Compute the predicted labels by selecting indices of the maximum valued components of the columns of $\mathbf{B}\boldsymbol{\alpha}$.
Compute the classification accuracy

## 6. Experiments and Evaluation

We performed experiments on face, object, scene, and action recognition using standard data-sets and compared the results with the state-of-the-are approaches in the dictionary and classifier learning domain such as Discriminative Bayesian Dictionary Learning (DBDL) [25], Joint Bayesian Discriminative Classifier (JBDC) [24], SRC [32], C-KSVD [16], D-KSVD [31], and FDDL [33]. We implemented the approach presented in JBDC [24] in Python and generated the results for comparison, as JBDC is also based upon Non-parametric Bayesian-based framework. However, we have presented reported results [24] of other approaches, seeing a big gap in their classification accuracy as compared to [24]. Gibbs sampling for the solution of our model is explained in Algorithm 1. We have abbreviated our approach by DBCL (Dictionary based Bayesian Classifier Learning). We have observed significant improvement in the classification accuracy as compared to other approaches. We present the detail and the outcomes of the experiments below.

### 6.1 Face recognition

For face recognition, we trained our model on Extended YaleB and AR database .

#### *6.1.1 Extended YaleB database*

This database was developed for 38 subjects with sufficient variations in illuminations and expressions of the subjects and contains 2,414 images. We used the dataset file provided by [16] containing 504-dimensional random face features extracted from 192 x 168 cropped face images [32]. We followed the protocol of experimentation of [24] i.e., training data contains randomly selected fifteen examples from each class, and the rest of the data is used as test data. We took the average of 10 experiments along with standard deviation to compare our results with other approaches. We report $\pm$std dev (% accuracy with standard deviation), average training time for all training samples, and average recognition time for a test sample in Table 1. Our approach of learning the same representations obviously affects the performance of the classifier and a significant improvement in the results is noted here. Training time reduction is attributed to training the same coefficients, as in the case of training different coefficients additional computation resources would increase the training time.

**Table 1.** Face recognition for Extended YaleB database . Note: Results are based upon 10 experiments and test time and train time are in milliseconds and minutes respectively

| Method | Accuracy(%) | Test Time | Train Time |
|---|---|---|---|
| LC-KSVD [16] | 89.73±0.59 | 0.60 | — |
| D-KSVD [31] | 89.77±0.57 | 0.61 | — |
| SRC [32] | 89.71±**0.45** | 50.19 | — |
| FDDL [33] | 90.01±0.69 | 42.82 | — |
| DBDL [25] | 91.09±0.59 | 1.07 | — |
| JBDC [24] | 92.32±0.70 | 0.18 | 30.10 |
| DBCL | **93.75**±0.76 | **0.17** | **29.00** |

*6.1.2 AR face database*

This database was developed by capturing 26 photographs each of 126 persons at two different times to induce variations in facial disguise, illuminations, and expressions. Consequently, this database consists of over 4,000 face images. We use 540-dimensional random-face features provided by [16] that were extracted by projecting 165 x 120 cropped face images onto 540-dimensional vectors using a random projection matrix. We performed an experiment on 2,600 images consisting of 50 male and 50 female subjects. The training set is formed by randomly selecting seven images from each subject and the rest of the images were used for testing. We observe the same trend of increase in accuracy and reduction in training time in the results listed in Table 2 for AR Face database. The average dictionary size came out to be 697.

**Table 2.** Face recognition for AR database. Note: The results are based on 10 experiments, and test time and train time are in milliseconds and minutes

| Method | Accuracy (%) | Test Time | Train Time |
|---|---|---|---|
| SRC [32] | 84.60±1.37 | 59.91 | — |
| LC-KSVD [16] | 85.37±1.34 | 0.91 | — |
| D-KSVD [31] | 85.41±1.49 | 0.92 | — |
| FDDL [33] | 85.97±1.23 | 50.03 | — |
| DBDL [25] | 86.15±1.19 | 1.20 | — |
| JBDC [24] | 88.90± 0.75 | 0.21 | 35.03 |
| DBCL | **89.91 ± 0.58** | **0.16** | **34.20** |

6.2 Object Classification

Caltech-101 database [36] involves 101 categories of objects and consists of 9,144 image samples along with a class of background images. The size of each class varies from 31 to 800. 4096-dimensional feature vectors were extracted from the data by training the 16-layer deep convolutional neural networks for large-scale visual recognition [37]. Our experiment consists of six stages in which we randomly selected 5, 10, 15, 20, 25, and 30 samples per class respectively for training data sets. The trend of our results continues here and the listed results in Table 3 clearly show significant improvement in accuracy as compared to other approaches.

**Table 3.** Object classification for Caltech-101 database with six stages consisting of randomly selected 5, 10, 15, 20, 25, and 30 data points from each class for training sets respectively. Note: Accuracy is in percentage and training time (Tr. Time) is in minutes

| **Training Samples→** | | **5** | **10** | **15** | **20** | **25** | **30** |
|---|---|---|---|---|---|---|---|
| SRC [32] | Accuracy | 76.23 | 79.99 | 81.27 | 83.48 | 84.00 | 84.51 |
| | Tr. Time | – | – | – | – | – | – |
| FDDL [33] | Accuracy | 78.31 | 81.37 | 83.37 | 84.76 | 85.66 | 85.98 |
| | Tr. Time | – | – | – | – | – | – |
| D-KSVD [31] | Accuracy | 79.69 | 83.11 | 84.99 | 86.01 | 86.80 | 87.72 |
| | Tr. Time | – | – | – | – | – | – |
| LC-KSVD [16] | Accuracy | 79.74 | 83.13 | 85.20 | 85.98 | 86.77 | 87.81 |
| | Tr. Time | – | – | – | – | – | – |
| DBDL [25] | Accuracy | 80.11 | 84.03 | 85.99 | 86.71 | 87.97 | 88.81 |
| | Tr. Time | – | – | – | – | – | – |
| JBDC [24] | Accuracy | 82.92 | 89.60 | 91.65 | 92.81 | 93.98 | 93.82 |
| | Tr. Time | 18.00 | 50.53 | 100.32 | 120.11 | 150.03 | 250.33 |
| DBCL | Accuracy | **83.89** | **90.28** | **92.39** | **93.20** | **94.69** | **94.83** |
| | Tr. Time | **11.12** | **35.12** | **80.57** | **100.43** | **133.86** | **200.00** |

6.3 Scene categorization

The Fifteen Scene Category database [38] involves fifteen natural scene categories and each image is of average size $250 \times 300$ of pixels. The number of samples for each category varies from $200$ to $400$. We used 3000-dimensional Spatial Pyramid Features of the samples provided by Jiang et al. [16]. For constructing the training data set, we randomly selected 50 samples from each category, and the remaining data was used for testing. We have reported the results of 10 experiments in Table 4. Our approach outperformed other approaches.

**Table 4.** Fifteen Scene Category database based on ten experiments. Note: Test Time and Train Time are in milli seconds and minutes

| **Method** | **Accuracy (%)** | **Test Time** | **Train Time** |
|---|---|---|---|
| FDDL [33] | 94.08±0.43 | 57.99 | — |
| D-KSVD [31] | 95.12±0.18 | 0.58 | — |
| LC-KSVD [16] | 95.37±0.28 | 0.59 | — |
| SRC [32] | 95.41±0.13 | 78.33 | — |
| DBDL [25] | 96.98 ± 0.28 | 0.71 | – |
| JBDC [24] | 97.45 ± 0.28 | 1.33 | 37.22 |
| DBCL | **98.10 ±0.08** | **0.66** | **30.11** |

6.4 Action recognition

We used action bank featured (processed data) [39] of UCF sports action database [40] for action recognition. The database consists of $150$ clips $@10fps$ taken for $10$ classes of varied sports actions. Our experiment consists of five Five-Fold group-wise cross-validations with different seed values for each

cross-validation. The model was trained and tested on 25 partitions and the mean recognition rates were reported in Table 5. The proposed approach outperformed as compared to other approaches.

**Table 5.** Action recognition for Action bank features (Processed data) of UCF Sports Action database with five Five-fold cross validations experiment. Note: Test Time and Train Time are in milliseconds and minutes

| Method | Accuracy (%) | Test Time | Train Time |
|---|---|---|---|
| JBDC [24] | 93.43 $\pm$ 4.37 | 15.44 | 30.32 |
| DBCL | **95.00 $\pm$1.75** | **4.98** | **15.00** |

**7. Discussion**

Prominent feature of our approach is training the same representation coefficients for training examples and the corresponding labels. Consequently, we gain the classification accuracy as compared to the previous approaches. The results support our claim of improvement in the accuracy. Additionally, our model also gives an additive advantage of improvement in training time in conjunction with the improvement in classification accuracy. Our approach frees computational resources that would have been engaged in learning different coefficients at classifier level, resulting in saving training time. We tuned the hyperparameters of our model in conjunction with the theoretical background mentioned in [24]. Accordingly, we conclude $0 < a_0, b_0 < \min_c |I_c|$. Mathematically, $N \to \infty$ for initialization of dictionary, but $K > N$ is large enough to serve the same purpose for initialization of the size of the dictionary and the classifier. Eventually, number of dictionary atoms decreases to less than N as the result of dictionary atoms pruning during iterations of the Gibbs sampler. Higher values of $\lambda_a$ and $\lambda_h$ induce precision in the data and label representation, suiting our data because most of the data is clean. As we saw in the case of UCF dataset, we set these values at $10^9$ due to the availability of a very small size of data. We observed that the value of $\lambda_s^c = 1$ remains almost equal to $1$ that was set to initialize this parameter. We also observed that values of $\lambda_h$ and $\lambda_a$ remain around initial values during training. However, we have observed convergence issues in case $\lambda_h$ and $\lambda_a$ are initialized with smaller values.

**8. Conclusion**

We used Gibbs sampler as inference algorithm to learn the posterior parameters of our model. The Gibbs sampling is a robust technique to solve probabilistic models mostly used by the majority of authors. Due to introduction of a new idea of learning same representation coefficients, the results are found to be improved as compared to other state-of-the-art approaches. Learning different representations for training examples and the corresponding labels, but using the representation of the test sample computed over the dictionary is not intuitively convincing. Therefore, the improvement in the results is attributed to the learning of the same representations for data samples and the corresponding labels. As an added advantage, we also achieved a gain in training time, because of the reduction of the computational cost by learning one set of coefficients for the representation of data and the corresponding labels at the dictionary and the classifier learning stages. Moreover, the formulation of the problem in Bayesian settings enables us to use the robust optimization algorithm i.e., Gibbs sampler for solving probabilistic Bayesian networks in an efficient way.

**Acknowledgment:** We thank all the researchers at "TUKL LAB" of the School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan who maintained an excellent research environment.

**Data and Code Availability:** The availability detail of the datasets we have used in our work is given below. The random features for the Extended YaleB database, AR face database, and fifteen scene category database are publically available at http://www.zhuolin.umiacs.io/projectlcksvd.html [16] and can be downloaded by clicking at http://www.zhuolin.umiacs.io/LCKSVD/features/randomfaces4extendedyaleb.zip, http://www.zhuolin.umiacs.io/LCKSVD/features/randomfaces4ar.zip, and http://www.zhuolin.umiacs.io/LCKSVD/features/spatialpyramidfeatures4scene15.zip respectively.

Processed data of the UCF sports database is publically available at the site https://cse.buffalo.edu/~jcorso/r/actionbank/#SaCoCVPR2012 and can be downloaded by clicking at the link http://www.cse.buffalo.edu/~jcorso/extdelivery/ab_ucfsports.tar.bz from the same site.

**References**


1. V. Naumova and K. Schnass, "Dictionary learning from incomplete data for efficient image restoration," in *2017 25th European signal processing conference (EUSIPCO)*, 2017, pp. 1425–1429. doi: 10.23919/EUSIPCO.2017.8081444.
2. P. Song and M. R. D. Rodrigues, "Multimodal image denoising based on coupled dictionary learning," in *2018 25th IEEE international conference on image processing (ICIP)*, 2018, pp. 515–519. doi: 10.1109/ICIP.2018.8451697.
3. J. Li, J. Wang, and J. Li, "Image denoising algorithm based on incoherent dictionary learning," in 2019 chinese control conference (CCC), 2019, pp. 3337–3340. doi: 10.23919/ChiCC.2019.8865193.
4. D. Wang, J. Chen, Q. Zhang, and J. Wan, "A novel online dictionary learning method from compressed signals," in 2016 8th international conference on intelligent human-machine systems and cybernetics (IHMSC), 2016, pp. 351–354. doi: 10.1109/IHMSC.2016.45.
5. F. Lin, Z. Fei, J. Wan, N. Wang, and D. Chen, "A robust efficient dictionary learning algorithm for compressive data gathering in wireless sensor networks," in 2017 9th international conference on intelligent human-machine systems and cybernetics (IHMSC), 2017, pp. 12–15. doi: 10.1109/IHMSC.2017.118.
6. Y. Ji, W.-P. Zhu, and B. Champagne, "Structured dictionary learning for compressive speech sensing," in 2018 26th european signal processing conference (EUSIPCO), 2018, pp. 573–577. doi: 10.23919/EUSIPCO.2018.8553551.
7. H. Foroughi, M. Shakeri, N. Ray, and H. Zhang, "Face recognition using multi-modal low-rank dictionary learning," in 2017 IEEE international conference on image processing (ICIP), 2017, pp. 1082–1086. doi: 10.1109/ICIP.2017.8296448.
8. J. Ge, T. Zhou, F. Zhang, and K. Tse, "Learning part-based dictionary by sparse NMF for face gender recognition," in 2015 8th international symposium on computational intelligence and design (ISCID), 2015, pp. 375–378. doi: 10.1109/ISCID.2015.149.
9. M. Yang, Q. Wang, W. Wen, and Z. Lai, "Multi-feature joint dictionary learning for face recognition," in 2017 4th IAPR asian conference on pattern recognition (ACPR), 2017, pp. 629–633. doi: 10.1109/ACPR.2017.138.
10. H. Moeini and S. Mozaffari, "Gender dictionary learning for gender classification," Journal of Visual Communication and Image Representation, vol. 42, pp. 1–13, 2017, doi: https://doi.org/10.1016/j.jvcir.2016.11.002.
11. Z. Pei, Y. Wang, and J. M. Afridon, "Dictionary pair learning in compressed space for action recognition," in 2017 IEEE 3rd information technology and mechatronics engineering conference (ITOEC), 2017, pp. 313–317. doi: 10.1109/ITOEC.2017.8122306.
12. L. Wang, Z. Liu, R. Wang, and H. Qi, "SSM-based joint dictionary learning for cross-view action recognition," in 2019 chinese control and decision conference (CCDC), 2019, pp. 1628–1632. doi: 10.1109/CCDC.2019.8833150.
13. J. Zhang, H. P. H. Shum, J. Han, and L. Shao, "Action recognition from arbitrary views using transferable dictionary learning," IEEE Transactions on Image Processing, vol. 27, no. 10, pp. 4709–4723, 2018, doi: 10.1109/TIP.2018.2836323.
14. A. Castrodad and G. Sapiro, "Sparse modeling of human actions from motion imagery," International Journal of Computer Vision, vol. 100, no. 1, pp. 1–15, 2012, doi: 10.1007/s11263-012-0534-7.
15. W. Li et al., "An efficient face classification method based on shared and class-specific dictionary learning," in 2015 IEEE international conference on image processing (ICIP), 2015, pp. 2596–2600. doi: 10.1109/ICIP.2015.7351272.
16. Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 11, pp. 2651–2664, 2013, doi: 10.1109/TPAMI.2013.88.
17. N. Zhou, Y. Shen, J. Peng, and J. Fan, "Learning inter-related visual dictionary for object recognition," in 2012 IEEE conference on computer vision and pattern recognition, 2012, pp. 3490–3497. doi: 10.1109/CVPR.2012.6248091.

18. [18] L. Yan, R. Zhu, Y. Liu, and N. Mo, "Class-specific dictionary based semi-supervised domain adaptation for land-cover classification of aerial images," in IGARSS 2019 - 2019 IEEE international geoscience and remote sensing symposium, 2019, pp. 720–723. doi: 10.1109/IGARSS.2019.8898116.
19. F. Pan, Z.-X. Zhang, B.-D. Liu, and J.-J. Xie, "Class-specific sparse principal component analysis for visual classification," IEEE Access, vol. 8, pp. 110033–110047, 2020, doi: 10.1109/ACCESS.2020.3001202.
20. M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," IEEE Trans. Image Processing, vol. 15, no. 12, pp. 3736–3745, 2006, doi: 10.1109/TIP.2006.881969.
21. M. Yang, L. Zhang, J. Y., and D. Zhang, "Metaface learning for sparse representation based face recognition," in Proceedings of the international conference on image processing, ICIP 2010, september 26-29, hong kong, china, IEEE, 2010, pp. 1601–1604. doi: 10.1109/ICIP.2010.5652363.
22. [22] J. Mairal, F. R. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in 2008 IEEE computer society conference on computer vision and pattern recognition (CVPR 2008), 24-26 june 2008, anchorage, alaska, USA, IEEE Computer Society, 2008. doi: 10.1109/CVPR.2008.4587652.
23. Qiang Qiu, Z. Jiang, and R. Chellappa, "Sparse dictionary-based representation and recognition of action attributes," in 2011 international conference on computer vision, Nov. 2011, pp. 707–714. doi: 10.1109/ICCV.2011.6126307.
24. N. Akhtar, A. Mian, and F. Porikli, "Joint discriminative bayesian dictionary and classifier learning," in 2017 IEEE conference on computer vision and pattern recognition (CVPR), 2017, pp. 3919–3928. doi: 10.1109/CVPR.2017.417.
25. N. Akhtar, F. Shafait, and A. Mian, "Discriminative bayesian dictionary learning for classification," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 12, pp. 2374–2388, 2016, doi: 10.1109/TPAMI.2016.2527652.
26. D.-S. Pham and S. Venkatesh, "Joint learning and dictionary construction for pattern recognition," in 2008 IEEE computer society conference on computer vision and pattern recognition (CVPR 2008), 24-26 june 2008, anchorage, alaska, USA, IEEE Computer Society, 2008. doi: 10.1109/CVPR.2008.4587408.
27. J. Yang, K. Yu, and T. S. Huang, "Supervised translation-invariant sparse coding," in The twenty-third IEEE conference on computer vision and pattern recognition, CVPR 2010, san francisco, CA, USA, 13-18 june 2010, IEEE Computer Society, 2010, pp. 3517–3524. doi: 10.1109/CVPR.2010.5539958.
28. J. W. Paisley and L. Carin, "Nonparametric factor analysis with beta process priors," in Proceedings of the 26th annual international conference on machine learning, ICML 2009, montreal, quebec, canada, june 14-18, 2009, A. P. Danyluk, L. Bottou, and M. L. Littman, Eds., in ACM international conference proceeding series, vol. 382. ACM, 2009, pp. 777–784. doi: 10.1145/1553374.1553474.
29. M. Zhou, H. Chen, L. Ren, G. Sapiro, L. Carin, and J. W. Paisley, "Non-parametric bayesian dictionary learning for sparse image representations," in Advances in neural information processing systems 22, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds., Curran Associates, Inc., 2009, pp. 2295–2303.
30. Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in Proceedings of 27th asilomar conference on signals, systems and computers, Nov. 1993, pp. 40–44. doi: 10.1109/ACSSC.1993.342465.
31. Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in The twenty-third IEEE conference on computer vision and pattern recognition, CVPR 2010, san francisco, CA, USA, 13-18 june 2010, IEEE Computer Society, 2010, pp. 2691–2698. doi: 10.1109/CVPR.2010.5539989.
32. J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 2, pp. 210–227, 2009, doi: 10.1109/TPAMI.2008.79.

33. M. Yang, L. Zhang, X. Feng, and D. Zhang, "Sparse representation based fisher discrimination dictionary learning for image classification," International Journal of Computer Vision, vol. 109, no. 3, pp. 209–232, 2014, doi: 10.1007/s11263-014-0722-8.
34. 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," IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 6, pp. 643–660, 2001, doi: 10.1109/34.927464.
35. A. Martinez and R.Benavente, "The RA face database." 1998.
36. F.-F. Li, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," Computer Vision and Image Understanding, vol. 106, no. 1, pp. 59–70, 2007, doi: 10.1016/j.cviu.2005.09.012.
37. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv 1409.1556, 2014.
38. S. Lazebnik, C. Schmid, and 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 2006), 17-22 june 2006, new york, NY, USA, IEEE Computer Society, 2006, pp. 2169–2178. doi: 10.1109/CVPR.2006.68.
39. S. Sadanand and J. J. Corso, "Action bank: A high-level representation of activity in video," in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2012.
40. M. D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH a spatio-temporal maximum average correlation height filter for action recognition," in 2008 IEEE conference on computer vision and pattern recognition, 2008, pp. 1–8. doi: 10.1109/CVPR.2008.4587727.