applied sciences

MDPI

Article

Sparse Representations Optimization with Coupled Bayesian Dictionary and Dictionary Classifier for Efficient Classification

Muhammad Riaz-ud-din [1,*], Salman Abdul Ghafoor [1] and Faisal Shafait [1,2]

[1] School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan; salman.ghafoor@seecs.edu.pk (S.A.G.); faisal.shafait@seecs.edu.pk (F.S.)
[2] National Center of Artificial Intelligence (NCAI), Islamabad 46000, Pakistan
* Correspondence: muhammad.riaz@seecs.edu.pk

Abstract: Among the numerous techniques followed to learn a linear classifier through the discriminative dictionary and sparse representations learning of signals, the techniques to learn a nonparametric Bayesian classifier jointly and discriminately with the dictionary and the corresponding sparse representations have drawn considerable attention from researchers. These techniques jointly learn two sets of sparse representations, one for the training samples over the dictionary and the other for the corresponding labels over the dictionary classifier. At the prediction stage, the representations of the test samples computed over the learned dictionary do not truly represent the corresponding labels, exposing weakness in the joint learning claim of these techniques. We mitigate this problem and strengthen the joint by learning a set of weights over the dictionary to represent the training data and further optimizing the same weights over the dictionary classifier to represent the labels of the corresponding classes of the training data. Now, at the prediction stage, the representation weights of the test samples computed over the learned dictionary also represent the labels of the corresponding classes of the test samples, resulting in the accurate reconstruction of the labels of the classes by the learned dictionary classifier. Overall, a reduction in the size of the Bayesian model's parameters also improves training time. We analytically and nonparametrically derived the posterior conditional probabilities of the model from the overall joint probability of the model using Bayes' theorem. We used the Gibbs sampler to solve the joint probability of the model using the derived conditional probabilities, which also supports our claim of efficient optimization of the coupled/joint dictionaries and the sparse representation parameters. We demonstrated the effectiveness of our approach through experiments on the standard datasets, i.e., the Extended YaleB and AR face databases for face recognition, Caltech-101 and Fifteen Scene Category databases for categorization, and UCF sports action database for action recognition. We compared the results with the state-of-the-art methods in the area. The classification accuracies, i.e., 93.25%, 89.27%, 94.81%, 98.10%, and 95.00%, of our approach on the datasets have increases of 0.5 to 2% on average. The overall average error margin of the confidence intervals in our approach is 0.24 compared with the second-best approach, JBDC, for which it is 0.34. The AUC–ROC scores of our approach are 0.98 and 0.992, which are better than those of others, i.e., 0.960 and 0.98, respectively. Our approach is also computationally efficient.

Keywords: linear classifier; dictionary learning; nonparametric Bayesian; discriminative; sparse representation

Citation: Riaz-ud-din, M.; Ghafoor, S.A.; Shafait, F. Sparse Representations Optimization with Coupled Bayesian Dictionary and Dictionary Classifier for Efficient Classification. *Appl. Sci.* **2024**, *14*, 306. https://doi.org/10.3390/app14010306

Academic Editor: Luigi Portinale

Received: 1 November 2023
Revised: 13 December 2023
Accepted: 20 December 2023
Published: 29 December 2023

1. Introduction

Dictionary learning and sparse representation are considerably used in the different approaches of research areas, particularly in image restoration [1–4], compressive sensing in ad hoc wireless sensors networks and image processing areas [3,5,6], face and gender classification [7–14], action recognition [15–18], face recognition and object detection [19–21], etc. A dictionary consists of a collection of column vectors trained in such a way that

examples of a dataset belonging to a particular domain can be expressed as linear combinations of a few column vectors of the collection. The column vectors are called atoms. For example, a sample taken from a particular class of dataset can be expressed as a sparse linear combination of the remaining samples of the same class. The atoms contributing to the linear combination to reconstruct a signal are few. Therefore, the weights associated with these atoms form a sparse representation of the signal. These weights are also called sparse weights.

The sparse representation power of a dictionary was first exploited by Wright et al. [9], proposing a sparse representation-based classification scheme (SRC). The test signal was sparsely represented over an overcomplete dictionary that was constructed by the labeled training signals. The test signal was constructed with the linear combination of dictionary atoms and the maximum count of class-specific atoms contributing to the linear combination decides the class of the test sample. Although the classification accuracy of SRC was good, computationally it was very expansive. Instead of using training data as a dictionary, discriminative dictionary-learning techniques were introduced for classification. These techniques learned discriminative dictionaries of reasonable sizes and resulted in improved accuracy and efficiency [12,22].

Instead of directly using labeled training examples to learn a linear classifier, sparse representations of these training examples were also used to learn linear classifiers separately. Joint classifier learning along with discriminative dictionary learning techniques became very effective and efficient [23–25]. The association of labels with the dictionaries and the classifiers makes the dictionary and the classifier discriminative. Discriminative learning enhances classification accuracy [19,24,25]. For a discriminative dictionary to be used for classification purposes, a dictionary may be divided into three categories. In the first category, it consists of two types of atoms, i.e., the class-specific atoms contributing to the representation of data samples belonging to a particular class and the atoms contributing to the representation of all samples of the data [8,26,27]. The second category consists of atoms grouped among classes, where each class of atoms represents the data belonging to that class only [3,14,28–30]. In the third category of dictionaries, a data sample is represented as a sparse linear combination of atoms selected dictionary-wide without grouping the atoms among classes. However, discrimination in the dictionary is induced during learning in such a way that sparse weights become discriminative and are used for classification [8,31]. These three types of discriminative features are equally applicable to a dictionary classifier, which either consists of a single dictionary or a classifier that has been learned in conjunction with a dictionary.

The authors' key objective remains to keep the relationship between data samples and the corresponding labels intact while inducing discriminative behavior. In conjunction with discriminative learning, Akhtar et al. [24,25] introduced approaches for joint learning of dictionary and a linear classifier based upon a nonparametric Bayesian framework coupled with beta-Bernoulli processes. These approaches also introduced the adaptive learning feature of the size of the dictionary and classifier, and the association of labels of the training examples with dictionary and classifier atoms to make them discriminative to enhance classification accuracy. Beta-Bernoulli processes enabled discriminative learning by associating training examples' labels with the dictionary and classifier atoms. The same Bernoulli distributions were learned for the selection of dictionary and classifier atoms to represent training examples and the corresponding labels, respectively. Consequently, the joint discriminative and adaptive learning ability of the model enhanced the classification efficiency. These approaches jointly learn two different sets of sparse weights to represent training examples and their corresponding class labels over the dictionary and the classifier, respectively. Two steps are followed to predict a test sample. In the first step, sparse weights of the test sample are computed over the learned dictionary, using the OMP (orthogonal matching pursuit) algorithm. In the second step, these sparse weights are used as inputs to the learned classifier to predict the label of the test sample. The classifier acts like a dictionary, and its label reconstruction ability determines its efficiency in predicting

the test sample. In the second step, it is expected that the dictionary classifier will precisely construct the corresponding class label of the test sample using these sparse weights computed in the first step. However, during training, different sparse weights were learned to represent the corresponding labels of the training examples. Therefore, the performance of the classifier will be decreased. The use of weights computed over the dictionary will not effectively allow the classifier to reconstruct the corresponding label correctly.

These approaches [24,25] enhanced discriminative behavior while learning a dictionary and linear classifier, along with a feature of adaptively learning the size of the dictionary. However, these approaches are not strongly effective in establishing a strong joint between the dictionary and the classifier, as their strategies are of different sparse weights learning, and choosing the sparse weights computed over the dictionary instead of over the classifier is not intuitively feasible. To mitigate this issue and enhance the coupling between the dictionary and the dictionary classifier, we introduce the same representation weights learning approach. Our approach learns the same sparse weights over the dictionary to represent the training examples and over the classifier to represent the corresponding training labels, respectively. At the prediction stage, the representation weights of a test sample, computed over the dictionary, will also serve as the representation weights of the corresponding label over the classifier. This way, the reconstruction of the corresponding label or mapping to the corresponding label by the dictionary classifier will be more accurate. Therefore, the sparse codes of a signal computed over the dictionary should either be used separately for classification [9] or, if these are to be integrated with the dictionary classifier in joint learning settings [24,25], the same sparse codes should be further learned/optimized over the dictionary classifier, instead of learning different codes over the dictionary classifier. We point this out as a research gap that needs to be addressed. By learning the same weights, we address this gap in the existing research. We tailor the sparse weights being learned over the dictionary, representing training examples, to further optimize for the representation of corresponding labels over the dictionary classifier, instead of learning another set of weights that are not accessible at the stage of prediction. This way, the sparse weights get truly integrated with the dictionary and the classifier in joint learning settings. Moreover, we introduced a new strategy in the Gibbs sampling learning technique to improve the training time. We tested during experimentation that the number of inner iterations can be reduced by processing atoms and associated model parameters in groups. The size of the group varies from database to database to avoid degradation of the accuracy of the results. We used conjugate priors and analytically and nonparametrically derived the conditional probability distributions of the posterior parameters of the proposed Bayesian network. We used these conditional probabilities during Gibbs sampling for iteratively taking samples for computing the posterior probabilities of the model.

The following are the salient contributions of our work:

1. Our approach learns jointly and discriminately the dictionary and the dictionary classifier with the same sparse weights over the dictionary to represent training examples and over the classifier to represent the corresponding labels of the training examples. The approach highlights the weakness in the joint learning of the existing approaches and proposes a true strong joint design in Bayesian settings with enhanced classification accuracy.
2. Our approach is computationally efficient, as it reduces the training time by decreasing the overall number of parameters of the model.
3. The reconstruction error for the training examples needs to be updated after the update of each dictionary atom and the associated parameters during Gibbs sampling. We demonstrated that dictionary atoms and the associated parameters can be updated in groups, and reconstruction errors can be updated following group updating without compromising classification accuracy. However, the maximum size of the group of atoms varies depending on the data. Following this strategy, training time was reduced by a factor of 64.

In our work, we present the literature review in Section 2 and explain the problem in Section 3. We present the formulation of our proposed model along with the derivation of conditional probabilities for posterior parameters for Gibbs sampling in Sections 4 and 5, respectively. Our work also includes details of datasets and their preprocessing, performance measures, experiments and comparative results, statistical analysis, discussion, and conclusion in Sections 7–14, respectively.

2. Literature Review

Yang et al. [13] first proposed a sparse representation classification (SRC) method for face recognition exploiting the discrimination capability of the dictionary. A competitive environment of effective dictionary learning problems evolved, and such problems became very challenging. Motivated by the sparse representation capability of a discriminative dictionary, various dictionary learning methods evolved, such as supervised dictionary learning [32], the kernel method [33], the proximal method [34], nonparametric Bayesian inference [25], and so on.

Various techniques of learning dictionaries could be categorized as multiple dictionary learning [27], compact dictionary learning [35], and discriminative dictionary learning [8,12,36]. Zhou et al. [27] trained multiple class-specific dictionaries using visual correlation within the objects. An incoherence term was introduced to the loss function for inducing class-specific discrimination between various class-specific dictionaries [37]. However, this technique was computationally expensive. To address the large-scale datasets, compact dictionary learning techniques evolved. An entropy minimization rule was utilized to pick the dictionary atoms for the representation of examples based upon a probabilistic model [38]. Additionally, the kernel regularization method was also employed for nonlinear dictionary atoms training, which reduced the dimensions of features [33]. However, this method could not induce effective discrimination in the learned dictionaries. To induce discrimination in the dictionaries, authors followed approaches to get label information induced into the loss function at the training stage. Logistic loss [39], quadratic loss [8], and hinge loss [40] are some of the examples of commonly used loss functions. Using label along with feature data [23], a discriminative method, K-SVD (D-KSVD), was proposed to learn the dictionary and linear classifier simultaneously. Jiang et al. [8,41] enhanced the dictionary's discrimination ability by introducing a label-consistent constraint. Yang et al. [12] proposed a structured dictionary scheme via Fisher discrimination criteria, utilizing both the reconstruction errors and the sparse codes for classification. The recognition accuracy was increased by developing a method based on low-rank representation to eliminate the correlation between different categories [42]. These optimization-based dictionary learning algorithms could not adaptively learn sparsity level and noise variance to produce the best results.

To overcome these limitations, dictionary learning problems were handled through Bayesian techniques. A nonparametric beta-Bernoulli process model was developed by [43] to learn dictionary atoms and sparse codes for image denoising and interpolation, which is capable of inferring the number of dictionary atoms from training data. Akhtar et al. [24,25], Akhtar and Mian [44] presented a joint discriminative dictionary and classifier learning algorithm for image classification using Gibbs sampling. A variational inference method was developed by [45] to train a dictionary for image denoising.

Deep learning methods were also gaining popularity in the computer vision field [46]. Convolutional neural networks (CNNs) were taking a lead [47]. For instance, a deep residual network [48] with a network depth of 100 was proposed for image classification. Huang et al. [49] proposed a densely connected CNN to strengthen feature propagation, increasing the classification accuracy. The multiple-layer architecture was the success of deep learning methods for superior approximation capacity. The total number of parameters involved in a deep model is very large. For the optimization of massive parameters, a large volume of data are required during training along with computational power. Therefore, for medium-scale datasets, dictionary-learning-based methods are suitable com-

pared with deep learning methods. Consequently, our article focuses on dictionary and sparse-representation-based classification methods.

Class-specific discriminative dictionary learning approaches produced the best classification accuracies, but heavy computation was involved during the training and classification of a test sample. Joint and discriminative learning of the dictionary and the classifier gained significant attention from the researchers. Traditional dictionary and classifier learning techniques could not adaptively optimize the model parameters, particularly sparsity and the size of the dictionary and the classifier. Nonparametric Bayesian approaches coupled with beta-Bernoulli processes became very famous for adaptively learning the parameters and inducing discrimination [24,25,44]. However, the emphasis and the success in controlling the accuracy through the discrimination-inducing power of beta-Bernoulli processes ignored an important aspect of learning the same representation weights for training examples and their corresponding labels. The sparse weights for the representation of the test sample and the corresponding label play a pivotal role in the classification process. The optimization of the same sparse weights over the dictionary and the classifier is intuitively an effective approach to enhance classification performance. The literature review is summarized in Table 1.

Table 1. Summary of the literature review.

	Authors	Method(s)	Datasets	Accuracy (%)	Train vs. Test Examples (%)—Other Conditions
Training Examples as Dictionary	Ramirez et al. [37]	Classification and clustering via dictionary learning with structured incoherence and shared features	MNIST	97.00	Size of dictionaries = 800
			USPS	98.00	Size of the dictionaries = 80
			ISOLET	98.50	
			Brodatz dataset	99.60	
	Yang et al. [13]	Robust sparse coding (RSC) by modeling the sparse coding as a sparsity-constrained robust regression problem	Extended Yale B	99.40	50/50—Acc. decreases with low-dimension features
			AR	96.00	50/50—Acc. decreases with low-dimension features
			Multi-PIE	97.80	35/65—Acc. with smile expression
Non-Bayesian Dictionary (Discriminative/Joint) Learning	Mairal et al. [32]	Supervised dictionary learning (SDL) with generative training (SDL-G) and with discriminative learning (SDL-D) along with linear (L) and bilinear (BL) decision functions	MNIST	98.96	86/14—with SDL-D L
			USPS	96.46	78/22—with SDL-D L
			Brodatz dataset	83.16	50/50—with SD-G BL
	Jiang et al. [8]	Label Consistent K-SVD: Learning a Discriminative Dictionary for Recognition (LC-KSVD) with two variants, LC-KSVD1 and LC-KSVD2, with discriminative power and with both the discriminative and the constructive powers	Extended YaleB	96.70	50/50—LC-KSVD2
			AR	97.80	78/22—LC-KSVD2
			Caltech101	73.60	34/66—LC-KSVD2
			Caltech256	34.32	38/62—LC-KSVD2
			15 Scene	92.90	34/66—LC-KSVD2
			UCF	95.70	Fivefold cross-validation—LC-KSVD2
	Liu et al. [33]	A kernel regularized nonlinear dictionary learning for sparse coding with stacked autoencoder (SAE) networks used to jointly learn low embedding of the data samples and a dictionary	USPS	99.00	71/29—Dimension = 120
			Extended Yale B	96.00	78/22—Dimension = 350
			COIL-20	97.00	70/30—Dimension = 250
			GTSRB	98.00	76/24—Dimension = 350

Table 1. *Cont.*

	Authors	Method(s)	Datasets	Accuracy (%)	Train vs. Test Examples (%)—Other Conditions
	Zhang et al. [21]	Twin-Incoherent Self-Expressive Locality-Adaptive Latent Dictionary Pair Learning for Classification (SLatDPL)	YaleB	98.2	50/50
			AR	98.60	77/23
			CMU PIE	95.30	24/76
			MIT CBCL	99.80	15 random split train/test—3240 examples (324 images per person)
			UMIST	93.40	10/90
			15Scene	98.80	33/67
			ETH80	98.30	15/85
			Caltech101	72.72	8/92
			Caltech256	78.90	13/87
Bayesian Dictionary Learning (Discriminative/Joint)	Akhtar et al. [25]	Discriminative Bayesian Dictionary Learning for Classification (DBDC). Nonparametric Bayesian framework with beta process	Extended Yale B	97.19	50/50
			AR	97.41	77/23
			Caltech-101	74.60	7/93
			15Scene	98.67	33/67
			UCF sports action	95.1	Fivefold cross-validation with four folds for training and one for testing
	Akhtar et al. [24]	Joint Discriminative Bayesian Dictionary and Classifier Learning (JBDC). Nonparametric Bayesian framework with beta process	Extended Yale B	92.14	24/76
			AR	87.17	27/73
			Caltech-101	89.59	7/93
			15Scene	97.73	17/83
			UCF sports action	95.7	Fivefold cross-validation with four folds for training and one for testing
	Akhtar and Mian [44]	Nonparametric Coupled Bayesian Dictionary and Classifier Learning for Hyperspectral Classification	Indian Pines	92.64	10/90—Avg. Acc. = 93.31%, Kappa = 0.917
			Salinas Image	92.75	1/99—Avg. Acc. = 96.45%, Kappa = 0.918
			Pavia University Image	91.30	10/90—Avg. Acc. = 87.99%, Kappa = 0.884

3. Problem Representation

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

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

where $\mathbf{A} \in R^{M \times N}$ is training data that contain C classes, i.e., $\mathbf{A}^1, \mathbf{A}^2 \ldots \mathbf{A}^c \ldots \mathbf{A}^C$, indexed in I_N. I_c represents a set containing the indices of the training examples belonging to the cth class, and equivalently, $\sum_{c=1}^{C} |I_c| = N$, where $|.|$ is the cardinality of a set. In Equation (1), $\boldsymbol{\alpha} \in R^{K \times N}$ is the matrix of sparse codes representing examples in $\mathbf{A}$. In terms of classes of examples, a class of data can be expressed as

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

where $\boldsymbol{\alpha}^c \in R^{K \times |I_c|}$ is the matrix consisting of sparse representation vectors of the examples belonging to $\mathbf{A}^c$. The constrained optimization problem of dictionary and sparse codes learning can be put in the form

$$< \mathbf{\Phi}, \boldsymbol{\alpha} >= \min_{\mathbf{\Phi}, \boldsymbol{\alpha}} ||\mathbf{A} - \mathbf{\Phi}\boldsymbol{\alpha}||_F^2 \quad s.t. \forall i, ||\boldsymbol{\alpha}_i||_p \leq t, \tag{3}$$

where $\boldsymbol{\alpha}_i \in R^K$ is the ith column of $\boldsymbol{\alpha}$ that represents the sparse codes of the ith example of data $\mathbf{A}$. The constant t controls the sparsity of the columns of the sparse codes matrix. The $||.||_F$ and $||.||_p$ represent the Frobenius norm and l_p-norm of a vector, respectively. The sparse codes learned during dictionary learning can be used for training the parameters of a linear classifier as below:

$$\mathbf{B} = \min_{\mathbf{B}} \sum_{i=1}^{N} \mathcal{L}\{\mathbf{h}_i, f(\boldsymbol{\alpha}_i, \mathbf{B})\} + \lambda ||\mathbf{B}||_F^2, \tag{4}$$

where $\mathbf{B} \in R^{C \times K}$, $\mathcal{L}$, $\mathbf{h}_i \in \{0,1\}^C$, λ, and $f(.)$ represent classifier, loss function, class label for $\mathbf{a}_i$, regularization constant, and predicted label for $\mathbf{a}_i$, respectively. Though the aforementioned approach provides a baseline for using the dictionary learning domain in the classification domain, this does not exhibit joint and discriminative behavior. The joint and discriminatory learning dictionary and classifier obtain the label information of the training data induced into the learning process, and the efficiency of the classifier increases [8,40,50]. Among such joint discriminative approaches, the approach followed by [24] became very prominent and knocked out other approaches. This approach used a Bayesian nonparametric framework with beta process [51]. The authors in [52] introduced the beta process for image restoration and compressive sensing. This concept was further exploited by Akhtar et al. [24,25] for object and scene classification, as well as face and action recognition.

Basically, Paisley and Carin [51] developed a beta process for nonparametric factor analysis. It is represented by $BP(a_0, b_0, \bar{\mathbf{h}}_0)$ with $a_0 > 0, b_0 > 0$, and $\bar{\mathbf{h}}_0$ as the base measure. The drawing of atoms in the dictionary from the base measure can be expressed as below:

$$\begin{aligned} \bar{\mathbf{h}} &= \sum_k \pi_k \delta_{\boldsymbol{\phi}_k}(\boldsymbol{\phi}),\ k \in K = \{1 \ldots, K\}, \\ \pi_k &\sim \text{Beta}(\pi_k | a_0/K, b_0(K-1)/K), \\ \boldsymbol{\phi} &\sim \bar{\mathbf{h}}_0, \end{aligned} \tag{5}$$

where $\delta_{\boldsymbol{\phi}_k}(\boldsymbol{\phi}) = 1$ for $\boldsymbol{\phi} = \boldsymbol{\phi}_k$ and 0 otherwise. $\bar{\mathbf{h}}$ is a vector of probabilities for the selection of atoms and its kth component represents the selection probability 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 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. Training data can sparsely be approximated as $\mathbf{A} \approx \boldsymbol{\phi}\mathbf{Z}$, where the number of nonzero elements in a column of $\mathbf{Z}$ depends upon the value of K. However, in case of $K \to \infty$, the number of nonzero elements is a draw from a Poisson $(\frac{a_0}{b_0})$ distribution [51]. This approach was used by [25] and learned a linear classifier independently. However, discrimination in the dictionary was induced by drawing different sets of $\mathbf{B}_r$s for each class of data. Joint learning was further induced by [24]. To keep data labels intact 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 weights for the representation of data samples and the corresponding labels at the dictionary learning and the classifier learning stages, respectively. While predicting a test sample, weights computed over the learned dictionary are used directly as input to the classifier. The different weights learned for the representation of labels during classifier learning get ignored at the prediction stage. We introduced an approach in which this issue is mitigated by learning the same weights at both the stages, i.e., at the dictionary learning and the classifier learning stages for the representation of both the data examples and the corresponding labels.

4. Formulation of Our Approach

We design a Bayesian network and follow a nonparametric approach for computing the conditional probabilities of the nodes. Motivated by the sparse representation power of

an overcomplete dictionary, we learn sparse representations of training examples by jointly learning two Bayesian dictionaries. The same sparse representations learned with the first dictionary, called the dictionary, are further optimized by the second dictionary, called the linear classifier, to represent the labels of the corresponding classes of the examples over the classifier. Furthermore, we also used the beta-Bernoulli processes to induce discrimination in the dictionary and the classifier. The Bayesian network of our approach is shown in Figure 1.

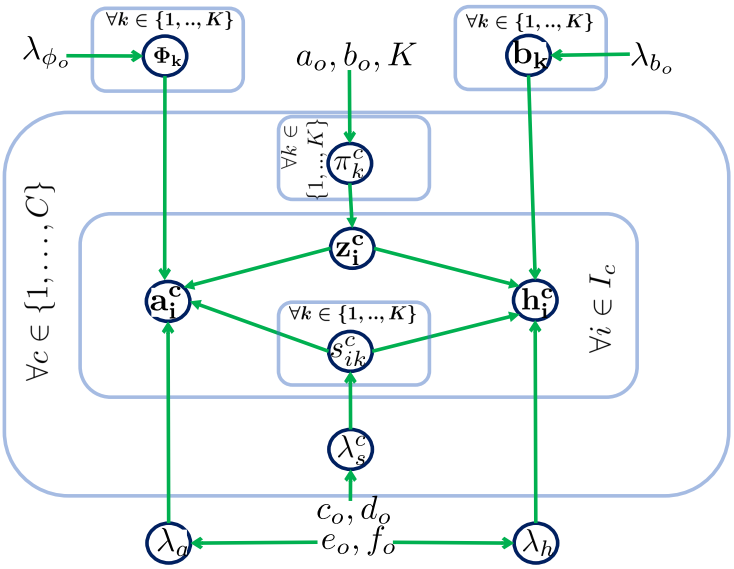

Figure 1. Bayesian Network. Nodes represent the random variables of the model and edges represent conditional dependence among the variables. Note that the same atom selection ($\mathbf{z}_i^c$) and representation (s_{ik}^c) variables contribute to the reconstruction of both the training examples, $\mathbf{a}_i^c$, and the labels, $\mathbf{h}_i^c$.

We use different base measures for the dictionary and the classifier and use the same Bernoulli distributions for the selection of the dictionary atoms and the classifier atoms for the representation of training examples and the corresponding labels. The Bernoulli distributions' parameters $\boldsymbol{\pi}$ are drawn from the beta process. These parameters are associated with the dictionary atoms and the classifier atoms. Though different base measures are used for beta-Bernoulli processes for the dictionary and the classifier, the same Bernoulli parameters are used for associating probabilities with dictionary atoms and the corresponding atoms of the classifier for the representation of the examples and the labels. However, to induce discrimination and associate labels with dictionary atoms and classifier atoms, we use different sets of draws of Bernoulli distribution parameters from beta-Bernoulli processes for each class of the training data. Consequently, different sets of Bernoulli distributions for the selection of atoms are used for the representation of the examples and the corresponding labels for different classes. The work in [24] uses two separate prior Gaussian distributions to draw the weights for the dictionary and the classifier for the representation of the training examples and the corresponding labels, respectively. We use the same Gaussian prior for drawing the same representation weights for the dictionary and the classifier to represent training examples and the corresponding labels. Based upon our nonparametric Bayesian approach for jointly learning the dictionary classifier with a dictionary and the same sparse representations for both the training examples and the labels of the corresponding classes, the model is expressed mathematically in Equation (6).

For the construction of the ith training example of the cth class. it is formulated as $\mathbf{a}_i^c = \boldsymbol{\Phi}\boldsymbol{\alpha}_i^c + \mathbf{a}_{\epsilon_i}$ and $\boldsymbol{\alpha}_i^c = \mathbf{z}_i^c \odot \mathbf{s}_i^c$. Here, $\boldsymbol{\alpha}_i^c$ is the sparse code vector (weights) for the ith example of data, $\mathbf{s}_i^c \in R^K$ is the weight vector associated with dictionary atoms contributing

to sparse code representation of ith example, and $\odot$ represents the Kronecker product. The linear classifier $\mathbf{B}$ is also learned jointly with the same sparse codes $\boldsymbol{\alpha}_i^c$ for the representation of class labels, contrary to [24]. The formal representation of our model is shown below.

$$\begin{aligned}
&\forall i \in I_c, \forall c \in \{1, 2, \ldots, C\}, \text{and } \forall k \in \{1, 2, \ldots, K\} \\
&\boldsymbol{\alpha}_i^c = \mathbf{z}_i^c \odot \mathbf{s}_i^c \\
&\mathbf{a}_i^c = \boldsymbol{\Phi}\boldsymbol{\alpha}_i^c + \mathbf{a}_{\epsilon_i} \quad \mathbf{h}_i^c = \mathbf{B}\boldsymbol{\alpha}_i^c + \mathbf{h}_{\epsilon_i} \\
&\pi_k^c \sim \text{Beta}(\pi_k^c | a_0/K, b_0(K-1)/K) \\
&z_{ik}^c \sim \text{Bernoulli}(z_{ik}^c | \pi_k^c) \\
&s_{ik}^c \sim \mathcal{N}(s_{ik}^c | 0, 1/\lambda_s^c) \\
&\boldsymbol{\phi}_k \sim \mathcal{N}(\boldsymbol{\phi}_k | \mathbf{0}, 1/\lambda_{\phi_0}\mathbf{I}_M) \quad \mathbf{b}_k \sim \mathcal{N}(\mathbf{b}_k | \mathbf{0}, 1/\lambda_{b_0}\mathbf{I}_C) \\
&\mathbf{a}_{\epsilon_i} \sim \mathcal{N}(\mathbf{a}_{\epsilon_i} | \mathbf{0}, 1/\lambda_a\mathbf{I}_M) \quad \mathbf{h}_{\epsilon_i} \sim \mathcal{N}(\mathbf{h}_{\epsilon_i} | \mathbf{0}, 1/\lambda_h\mathbf{I}_C)
\end{aligned} \tag{6}$$

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 kth 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. λ's are precision parameters of Gaussian priors in Equation (6). Bernoulli parameters π_i^c are drawn from the Beta distribution. To represent the kth column of the dictionary $\boldsymbol{\Phi}$ and the classifier $\mathbf{B}$, we use the notations $\boldsymbol{\phi}_k$ and $\mathbf{b}_k$, respectively, where $\mathbf{0}$ is the zero vector of dimension M for the dictionary prior and of dimension C for classifier prior. The subscript '0' appearing in the expressions shows the hyperparameters belonging to prior distributions. We also modeled errors $\mathbf{a}_{\epsilon_i}$ and $\mathbf{h}_{\epsilon_i}$ for the construction of both $\mathbf{a}_i^c$ and $\mathbf{h}_i^c$. We further place noninformative Gamma hyperpriors over precision parameters, i.e., $\lambda_s^c \sim \text{Gam}(c_0, d_0)$ and $\lambda_a, \lambda_h \sim \text{Gam}(e_0, f_0)$.

In conjunction with the priors defined in Equation (6) and our model shown in Figure 1, we derive conditional probabilities of posterior parameters $(\boldsymbol{\phi}_k, \mathbf{b}_k, s_{ik}^c, \boldsymbol{\pi}_k^c, \mathbf{z}_i^c)$ of our model in the coming sections.

5. Posterior Conditional Probabilities for Gibbs Sampling

Using the Gibbs sampler as an inference algorithm, we iteratively take samples from conditional probabilities for the posterior parameters of our model. We drive conditional probabilities analytically using conjugate priors of our proposed probabilistic model presented in Equation (6). We derived the factorized expressions for these conditional probabilities from the overall joint probability distribution of our model using the Bayes theorem. The symbol "$|-$" in the following conditional probabilities of the posterior variables means conditioned on all variables except the variable of the mentioned probability. Here, it is understood that the conditional probability is conditionally independent of all the variables absent in the expression, i.e., the variables outside the Markov blanket. This can be inferred from the probabilistic graphical model (PGM) presented in Figure 1. The overall joint probability of the model is given below.

$$\begin{aligned}
&p(\boldsymbol{\phi}, \mathbf{B}, \mathbf{A}, \mathbf{H}, \mathbf{Z}, \mathbf{S}, \boldsymbol{\lambda}_s, \boldsymbol{\pi}, \lambda_a, \lambda_h) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{\phi}_k | \mathbf{0}, \lambda_{\phi_0}^{-1}\mathbf{I}_M) \prod_{k=1}^{K} \mathcal{N}(\mathbf{b}_k | \mathbf{0}, \lambda_{b_0}^{-1}\mathbf{I}_C) \\
&\prod_{i=1}^{N}\prod_{k=1}^{K} \mathcal{N}(\mathbf{a}_{i_{\phi_k}} | \boldsymbol{\phi}_k(z_{ik}.s_{ik}), \lambda_a^{-1}\mathbf{I}_M) \prod_{i=1}^{N}\prod_{k=1}^{K} \mathcal{N}(\mathbf{h}_{i_{b_k}} | \mathbf{b}_k(z_{ik}.s_{ik}), \lambda_h^{-1}\mathbf{I}_C) \prod_{c=1}^{C}\prod_{i \in I_c}\prod_{k=1}^{K} \text{Bernoulli}(z_{ik}^c | \pi_k^c) \\
&\prod_{c=1}^{C}\prod_{i \in I_c}\prod_{k=1}^{K} \mathcal{N}(s_{ik}^c | 0, (\lambda_s^c)^{-1}) \prod_{c=1}^{C} \text{Gam}(\lambda_s^c | c_0, d_0) \prod_{c=1}^{C}\prod_{k=1}^{K} \text{Beta}\left(\pi_k^c | \frac{a_0}{K}, \frac{b_0(K-1)}{K}\right) \text{Gam}(\lambda_a | e_0, f_0) \\
&\text{Gam}(\lambda_h | e_0, f_0)
\end{aligned} \tag{7}$$

Bayes's theorem, in general, is given as

$$p(\Theta|X) = \frac{p(\Theta \bigcap X)}{p(X)}$$
$$\text{or} \tag{8}$$
$$p(\Theta|X) \propto p(\Theta \bigcap X) = p(X|\Theta)p(\Theta),$$

where $p(X)$ is the joint probability of observed data that is constant. In the following sections, we set up the factorized forms of the conditional probabilities of the model parameters and transform them analytically and nonparametrically to the standard forms of Gaussian, gamma, beta, and Bernoulli distributions belonging to the respective conjugate priors families.

5.1. Conditional Probability for Dictionary Atom ($\boldsymbol{\phi}_k$)

The factorized form of the conditional distribution of an atom of the dictionary can be written as

$$p(\boldsymbol{\phi}_k|-) \propto \prod_{i=1}^{N} \mathcal{N}(\mathbf{a}_{i_{\phi_k}}|\boldsymbol{\phi}_k(z_{ik}.s_{ik}), \lambda_a^{-1}\mathbf{I}_M)\mathcal{N}(\boldsymbol{\phi}_k|\mathbf{0}, \lambda_{\phi_0}^{-1}\mathbf{I}_M) \tag{9}$$

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 the reconstruction error induced by all dictionary atoms except the kth atom in representing $\mathbf{a}_i$. Here, the 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. The expression in Equation (9) can be analytically transformed to the standard Gaussian distribution as follows:

$$\boldsymbol{\phi}_k \propto \mathcal{N}(\boldsymbol{\phi}_k|\boldsymbol{\mu}_k, \lambda_\phi^{-1}\mathbf{I}_M), \quad \lambda_\phi = \lambda_{\phi_0} + \lambda_a \sum_{i=1}^{N}(z_{ik}.s_{ik})^2, \; \boldsymbol{\mu}_k = \lambda_a \lambda_\phi^{-1} \sum_{i=i}^{N}(z_{ik}.s_{ik})\mathbf{a}_{i_{\phi_k}} \tag{10}$$

5.2. Conditional Probability for Classifier Atom ($\mathbf{b}_k$)

Similarly,

$$\mathbf{b}_k \propto \mathcal{N}(\mathbf{b}_k|\boldsymbol{\mu}_k, \lambda_b^{-1}\mathbf{I}_C), \quad \lambda_b = \lambda_{b_0} + \lambda_h \sum_{i=1}^{N}(z_{ik}.s_{ik})^2, \; \boldsymbol{\mu}_k = \lambda_h \lambda_b^{-1} \sum_{i=i}^{N}(z_{ik}.s_{ik})\mathbf{h}_{i_{b_k}} \tag{11}$$

Here, $\mathbf{h}_{i_{b_k}}$ is the reconstruction error induced by all classifier atoms except the kth atom in representing $\mathbf{h}_i$. It may be noted here that we use the same weights, $\mathbf{s}_{ik}$, for both the dictionary and the classifier learning.

5.3. Conditional Probability for Bernoulli Variable (Z_{Ik}^C)

We derive the conditional probabilities of the posterior Bernoulli variables of our model. We use the same Bernoulli probabilities for the selection of both the dictionary atoms and the classifier atoms to represent training examples and the corresponding labels. The factorized form of the conditional probability for the posterior parameter z_{ik}^c is

$$p(z_{ik}^c|-) \infty \mathcal{N}(\mathbf{a}_{i_{\phi_k}}^c|\boldsymbol{\phi}_k(z_{ik}^c.s_{ik}^c), \lambda_a^{-1}\mathbf{I}_M)\mathcal{N}(\mathbf{h}_{i_{b_k}}|\mathbf{b}_k(z_{ik}^c.s_{ik}^c), \lambda_h^{-1}\mathbf{I}_C)\text{Bernoulli}(z_{ik}^c|\pi_k^c) \tag{12}$$

Equation (12) can be analytically transformed to the standard Bernoulli distribution, i.e., belonging to conjugate prior's family as follows:

$$\begin{aligned} &z_{ik}^c \sim \text{Bernoulli}\left(\frac{\pi_k^c \zeta_1 \zeta_2}{1 - \pi_k^c + \zeta_1 \zeta_2 \pi_k^c}\right), \;\; \zeta_1 = \exp\left(-\frac{\lambda_a}{2}(\boldsymbol{\phi}_k^T \boldsymbol{\phi}_k {s_{ik}^c}^2 - 2s_{ik}^c(\mathbf{a}_{i_{\phi_k}}^c)^T \boldsymbol{\phi}_k)\right), \\ &\zeta_2 = \exp\left(-\frac{\lambda_h}{2}(\mathbf{b}_k^T \mathbf{b}_k {s_{ik}^c}^2 - 2s_{ik}^c(\mathbf{h}_{i_{b_k}}^c)^T \mathbf{b}_k)\right) \end{aligned} \tag{13}$$

5.4. Conditional Probability for Representation Weight (s_{ik}^c)

This weight is associated with the kth atoms of the dictionary and the classifier, i.e., $\boldsymbol{\phi}_k$ and $\mathbf{b}_k$, during the linear combination of the dictionary atoms and the classifier atoms to construct the ith training example and the corresponding label, respectively. The factorized form of the conditional probability for representation weight is

$$p(s_{ik}^c|-) \propto \mathcal{N}(\mathbf{a}_{i_{\phi_k}}^c|\boldsymbol{\phi_k}(z_{ik}^c.s_{ik}^c), \lambda_a^{-1}\mathbf{I}_M)\mathcal{N}(\mathbf{h}_{i_{b_k}}^c|\mathbf{b}_k(z_{ik}^c.s_{ik}^c), \lambda_h^{-1}\mathbf{I}_C)N(s_{ik}^c|0, 1/\lambda_s^c), \tag{14}$$

Here, it may be noted that we are using the same Gaussian prior for representation weights with both the dictionary and the classifier. Equation (14) can be analytically transformed to the standard Gaussian distribution belonging to the conjugate prior's family as follows:

$$\begin{aligned} &s_{ik}^c \sim \mathcal{N}\left(s_{ik}^c|\mu_s, \lambda^{-1}\right), \;\; \lambda = \lambda_s^c + \lambda_a {z_{ik}^c}^2 \boldsymbol{\phi}_k^T\boldsymbol{\phi}_k + \lambda_h {z_{ik}^c}^2 \mathbf{b}_k^T\mathbf{b}_k, \\ &\mu_s = \lambda^{-1}\left(\lambda_a z_{ik}^c \boldsymbol{\phi}_k^T \mathbf{a}_{i_{\phi_k}}^c + \lambda_h z_{ik}^c \mathbf{b}_k^T \mathbf{h}_{i_{b_k}}^c\right) \end{aligned} \tag{15}$$

5.5. Conditional Probability for a Bernoulli Distribution Parameter (π_k^c)

The model parameter π_k^c is associated with the kth atoms of the dictionary and the classifier to be used as prior for the Bernoulli distribution for the selection of these atoms for the representation of training examples and the corresponding labels belonging to class c. The class-specific learning of the parameter induces discrimination in the dictionary and the classifier for associating labels with the dictionary and the classifier atoms. The factorized and standard forms of the conditional distribution of this parameter are

$$\begin{aligned} &p(\pi_k^c|-) \propto \prod_{i\in I_c} \text{Bernoulli}(z_{ik}^c|\pi_k^c)\text{Beta}\left(\pi_k^c|\frac{a_0}{K}, \frac{b_0(K-1)}{K}\right), \text{ or} \\ &p(\pi_k^c|-) \propto \text{Beta}\left(\frac{a_0}{K} + \sum_{i=1}^{|I_c|} z_{ik}^c, \frac{b_0(K-1)}{K} + |I_c| - \sum_{i=1}^{|I_c|} z_{ik}^c\right) \end{aligned} \tag{16}$$

As per analytical inference by [24], a dictionary atom ϕ_k can be pruned at each iteration of Gibbs sampling according to whether $\sum_{c=1}^{C} \boldsymbol{\pi}_k^c \to 0$ or not. Likewise, the classifier atom $\mathbf{b}_k$ is also pruned.

5.6. Conditional Distribution for Precision Parameter for a Representation Weight (λ_s^c)

A separate precision parameter for each class is learned to induce class-wise discrimination in the representation weights. The factorized form of the distribution is

$$p(\lambda_s^c|-) \propto \prod_{i\in I_c} \mathcal{N}(\mathbf{s}_i^c|\mathbf{0}, 1/\lambda_s^c\mathbf{I}_K)\text{Gam}(\lambda_s^c|c_0, d_0) \tag{17}$$

The expression is simplified as follows:

$$\lambda_s^c \sim \text{Gam}\left(\frac{|I_c|K}{2} + c_0, \frac{1}{2}\sum_{i=1}^{|I_c|}\|s_i^c\|_2^2 + d_0\right) \tag{18}$$

5.7. Conditional Probability for Precision Parameter for Data (λ_a)

The factorized form of the distribution is given as

$$p(\lambda_a|-) \propto \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{a}_i|\boldsymbol{\Phi}(\mathbf{z}_i \odot \mathbf{s}_i), \lambda_a^{-1}\mathbf{I}_M\right)\text{Gam}(\lambda_a|e_0, f_0) \tag{19}$$

The expression is analytically solved into the standard form of conjugate prior family as

$$\lambda_a \sim \text{Gam}\left(\frac{MN}{2} + e_0, \frac{1}{2}\sum_{i=1}^{N}\|\mathbf{a}_i - \mathbf{\Phi}(\mathbf{z}_i \odot \mathbf{s}_i)\|_2^2 + f_0\right) \tag{20}$$

5.8. Conditional Probability for Precision Parameter of Labels λ_h

Similarly,

$$\lambda_h \sim \text{Gam}\left(\frac{CN}{2} + e_0, \frac{1}{2}\sum_{i=1}^{N}\|\mathbf{h_i} - \mathbf{B}(\mathbf{z_i} \odot \mathbf{s_i})\|_2^2 + f_0\right) \tag{21}$$

After sufficient iterations of the Gibbs sampler, we compute the posterior probability distributions of dictionary atoms and the classifier parameters. For the prediction of a test sample, its sparse code representation $\boldsymbol{\alpha}$ over $\mathbf{\Phi}$ is computed first. The label of the test sample is predicted by classifying $\boldsymbol{\alpha}$ with the classifier $\mathbf{B}$. A label vector 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. Orthogonal matching pursuit (OMP) [53] 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.

6. Parameters Initialization

The overcomplete dictionary is initialized with a sufficiently large number of training samples on the order of 1.25 times the data. The data samples are randomly selected from the training data with replacement. We use OMP to compute sparse codes for the initialization of $\mathbf{s}_i^c$. 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 these are set equal to zero. The ridge regression technique is used to initialize the classifier $\mathbf{B}$, using $\mathbf{s}_i^c$ and training labels $\mathbf{h}_i^c \in R^C$ [8,23,25]. We set all π_k values equal to 0.5 to make the selection of dictionary and classifier atoms equally probable for the representation of data samples and the corresponding labels. We follow Algorithm 1 for Gibbs sampling.

Algorithm 1 Gibbs sampling

Require: We refer to Figure 1, Equation (6), and conditional probabilities derivations for this algorithm. Initialize the hyperparameters a_0, b_0 with $0 < a_0, b_0 < \min_c |I_c|$, c_0, d_0, e_0, f_0 with 10^{-6}, λ_{ϕ_0}, λ_{b_0} with M and C, λ_s^c with 1, and λ_a and λ_h with 10^9.
Initialize $\mathbf{\Phi}$, $\mathbf{B}$, $\boldsymbol{\pi}_k^c$, $\mathbf{z}_k^c$, and $\mathbf{s}_k^c$ as already described in Section 6.

for $i \in \{1, 2, 3, \ldots, 500\}$ **do**
  We can reduce the number of iterations in the inner loop by processing atoms in groups along with the associated parameters.
  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 $\boldsymbol{\pi}_k^c$ (using expressions of conditional distributions) $\forall c \in \{1, 2, 3, \ldots, C\}$
    $k = k + 1$
  end for
  Sample λ_a, λ_h, and λ_s^c ($\forall c \in \{1, 2, 3, \ldots, C\}$)
  $i = i + 1$
end for
Compute sparse weights $\boldsymbol{\alpha}$ of test data over the learned dictionary $\mathbf{\Phi}$, using orthogonal matching pursuit (OMP) available in the SPAMS package in Python. Compute predicted labels by selecting indices of the maximum value components of each column of $\mathbf{B}\boldsymbol{\alpha}$.
Compute the classification accuracy

7. Datasets and Preprocessing

In this research, two face recognition, two categorization, and one action recognition dataset are employed as benchmarks for experiments. We give the details of these datasets along with the preprocessing operations as follows.

7.1. Extended YaleB

This database was developed for 38 subjects, each with about 64 samples, with sufficient variations in illuminations and facial expressions containing 2414 face images [54]. These variations carried by each subject set a challenging stage for classification approaches. For one of the subjects, these variations are shown in Figure 2. We used 504-dimensional random face features [8] extracted by projecting 192×168 cropped face images [9] on a 504-dimensional vector. The projection matrix was generated from random samples of standard normal distributions for this transformation.

Figure 2. Face images for one subject from the YaleB dataset.

7.2. Ar Face Dataset

This database was developed by capturing 26 photographs of each of 126 subjects during two different sessions with larger variations in facial disguise, illumination, and expressions compared with YaleB [38]. Consequently, this database consists of over 4000 face images. Samples from this dataset are shown in Figure 3. We used 540-dimensional random face features that were extracted by projecting 165×120 cropped face images onto a 540-dimensional vector using a random projection matrix, as in Section 7.1.

Figure 3. Face images for one subject from the AR dataset.

7.3. Caltech-101 Dataset

The Caltech-101 database [55] consists of 101 categories of objects and comprises 9144 image samples along with a class of background images. It can be observed in Figure 4 that the images within each class have significant shape variations, setting a challenge for classification approaches. The size of each class varied from 31 to 800, and 4096-dimensional feature vectors were extracted from the data by training the 16-layer deep convolutional neural networks for large-scale visual recognition [56].

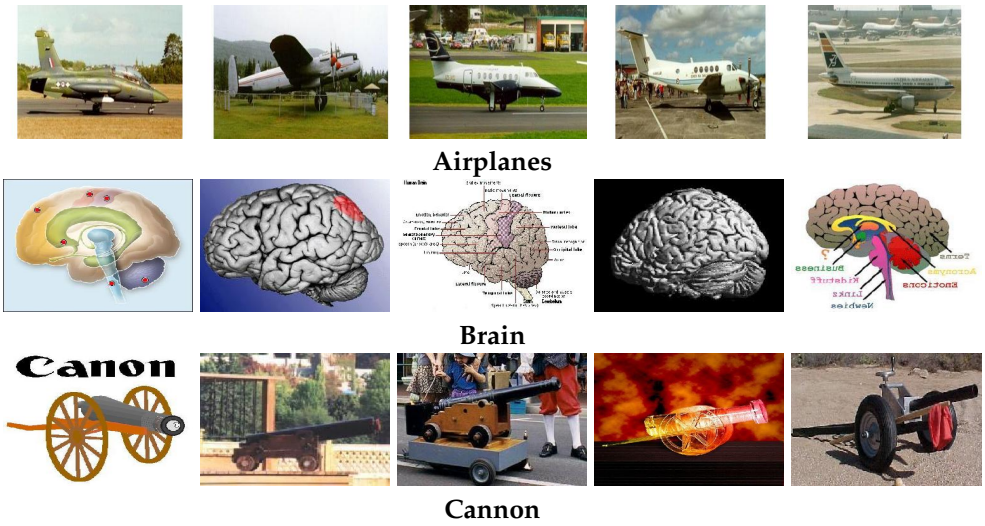

Figure 4. Images from three classes of the Catech-101 dataset.

7.4. Fifteen Scene Category Database

The Fifteen Scene Category database [57] involves fifteen natural scene categories, and each image has an average size of 250 × 300 pixels. The scenes include images from kitchens, living rooms, countrysides, etc. The number of samples for each category varies from 200 to 400. Figure 5 shows the images of eight of the fourteen categories. We used 3000-dimensional spatial pyramid features of the samples provided by [8].

Figure 5. Fifteen scene images; images from eight categories.

7.5. Sports Actions Database

We used action bank features (processed data) [58] of the UCF sports action database [59] for action recognition. The database consists of 150 clips @10fps taken for 10 classes of varied sports actions. The clips contain the sports actions that include kicking, golfing, diving, horse riding, skateboarding, running, swinging, swinging highbar, lifting, and walking. Figure 6 shows some of the action examples. Our experiment consists of five five-fold group-wise cross-validations with different seed values for each cross-validation. Four folds are used in training, and the remaining one is used for testing.

Figure 6. Action images from UCF actions dataset clips.

8. Performance Measures

We use the following performance metrics to evaluate the performance of our approach to establish a comparison with previous approaches. These metrics are explained below.

1. **Classification accuracy:** Classification accuracy is defined as

$$Acc.(\%) = (\frac{N_t}{N})100, \tag{22}$$

here, N_t and N are the number of test examples truly classified and the total number of test examples. We report the average accuracy, averaged over a number of experiments.

2. **Standard deviation:** We also evaluate our approach based on the standard deviation for an assessment of the error margins of the results. The standard deviation of the accuracy is measured as below.

$$\sigma = \sqrt{\frac{\sum (X - \mu)^2}{N}} \tag{23}$$

Here, σ, X, μ, and N represent the standard deviation, each value of the parameter, the mean value of the parameter, and the total number of experiments.

3. **Confidence interval:** We also measure confidence intervals based upon the confidence level of 95% for the classification accuracies. The details for finding confidence intervals based on a 95% confidence level are given in Section 10.
4. **AUC–ROC score:** The ROC (receiver operating characteristic) curve of a class is drawn based upon One-vs-the-Rest (OvR) settings, considering the class as positive and all others as negative. The curve shows the trend of true positive rate with false positive rate for different threshold values for the positive–negative settings. The smaller values of false positive rates and larger values of true positive rates favor the performance of the model for classifying the class. A higher AUC (area under the curve) value of the ROC of a class shows the good performance of the classifier in distinguishing between true and false predictions. The average area under the ROC curves of all the positive–negative combinations (all classes) determines the performance of the classification model. This value is called the AUC–ROC score.
5. **Training time:** The total time taken by an experiment during training of a model determines the computational efficiency of the model.
6. **Test time of prediction of a test example:** This time also shows the efficiency of the model in predicting a test example. In our approach, the sparse weights of a test sample over the learned dictionary are computed using orthogonal matching pursuit (OMP). Efficient optimization of the atoms of the dictionary determines how quickly OMP computes the sparse weights. Accordingly, the test time is either large or small, depending on the efficiency of the learned dictionary.
7. **Critical model parameters:** A few of the model parameters, like dictionary size and sparsity, along with other parameters, also play a crucial role in determining the efficiency of the model.

9. Experiments

We performed experiments for face recognition, object and scene categorization, and action recognition on standard datasets, i.e., the Extended YaleB [54] and AR [38] datasets, the Caltech-101 [55] and Fifteen Scene Category [57] datasets, and the UCF sports actions dataset [59], and we compared the results with the state-of-the-art methods, discriminative Bayesian dictionary learning (DBDL) [25], the joint Bayesian discriminative classifier (JBDC) [24], the sparse-representation-based classification (SRC) [9], the label-consistent K-SVD (LC-KSVD) [8], the discriminative K-SVD (D-KSVD) [23], the fisher discrimination dictionary learning (FDDL) [12], the joint analysis discriminative dictionary learning (ADDL) [19], the locality-constrained projective dictionary learning (LC-PDL) [20], and the twin-incoherent self-expressive latent dictionary pair learning (SLatDPL) [21]. These are the state-of-the-art approaches in the area of discriminative dictionary learning/sparse-representation-based classification. We reported the results for these approaches with the same protocols of experimentation followed for our approach.

In conjunction with reporting the classification accuracy, standard deviation, and training and testing times for all these methods in general, we also carried out a detailed statistical analysis of our approach with the nearest Bayesian approach (JBDC). In this analysis, we computed confidence intervals, computational efficiency, and AUC/ROC scores to establish the effectiveness of our approach. We report the outcomes of this analysis in Section 10.

In our approach, we set a_0 and $b_0 = min_c|I_c|$, $\forall c \in \{1, 2, \ldots, C\}$, and $K = 1.25N$ [24]. We set noninformative hyperparameters $c_0, d_0, e_0,$ and $f_0 = 10^{-6}$. We initialized precision parameters, i.e., λ_a and λ_h, with the value 10^9, except for the UCF sports dataset, where we used the value 10^{12} due to the small size of the training data. We initialized λ_s^c with 1 and set the values M and C for λ_{ϕ_0} and λ_{b_0}. A discussion on the parameter values selection of the proposed approach is provided in Section 13. We performed the experiments on an Intel Core i5 Processor with 16 GB RAM.

The results of the experiments for face recognition, object and scene classification, and action recognition experiments are discussed in the following sections.

9.1. Face Recognition for Extended Yaleb Database

We randomly selected fifteen examples from each class, and the rest of the data were used as test data. We conducted 10 experiments by randomly selecting the training and testing samples in each experiment. We report mean recognition accuracy, mean standard deviation, and test time in Table 2. The accuracy of our approach, i.e., 93.25, is the highest among all the listed methods. The improvement in the accuracy is attributed to the use of the same representations for learning of dictionary and the classifier. Improvement in the training time is attributed to a reduction in the overall number of the model parameters due to learning only a single set of representations. It is also noted that on using the dictionary pruning option, the dictionary size is adaptively reduced to 570 atoms starting from 712 atoms, i.e., 1.25 times the size of the training data, without degrading the accuracy. However, by fixing the dictionary size, we can further reduce the size to 564 atoms without compromising the accuracy. These dictionary sizes are less than the sizes of the dictionaries for other approaches, except for LC-KSVD and D-KSVD. We initialized the sparse codes with a sparsity of 25 computed over the initial dictionary, and the same sparsity value was used for the predicted labels. Decreasing the sparsity value from this value degraded the classification accuracy and increasing the sparsity value resulted in no gain in the accuracy. The reduction in the classification time per test sample is the result of the reduction in the dictionary size and the efficiency of the learned Bayesian dictionary in computing the sparse codes of a test sample over the dictionary. At the prediction stage, we use the orthogonal matching pursuit (OMP) algorithm [60] for computing the sparse codes over the learned dictionary. Further insight into the performance of the method is also presented in Section 10.

Table 2. Face recognition for Extended YaleB database [54]. Results are based on 10 experiments.

Method	Accuracy (%)	Test Time (ms)
LC-KSVD [8]	89.73 ± 0.59	0.60
D-KSVD [23]	89.77 ± 0.57	0.61
SRC [9]	89.71 ± **0.45**	50.19
FDDL [12]	90.01 ± 0.69	42.82
ADDL [19]	80.88 ± 1.07	—
LC-PDL [20]	80.90 ± 1.06	—
SLatDPL [21]	91.90 ± 0.72	—
DBDL [25]	91.09 ± 0.59	1.07
JBDC [24]	92.32 ± 0.64	**0.16**
Training Time (min)	38.40	
Our Method	**93.25** ± 0.72	**0.16**
Training Time (min)	**36.60**	

9.2. Face Recognition for Ar Face Database

We performed two sets of experiments on 2600 images of 50 male and 50 female subjects, each with 10 experiments. In the first set of experiments, the training sets were formed by randomly selecting seven images from each subject, and the rest were used for testing. In the second set of 10 experiments, training sets were formed by randomly selecting 20 examples from each subject, while taking the rest of the examples as test sets. We observe the same trend of increases in classification accuracy and reduction in training time, as in the case of the face recognition results for the YaleB dataset. The results are recorded in Table 3 for the AR face database. Classification accuracies of our approach in both sets of experiments are 89.27 and 98.20, respectively, which beat those of all other approaches. Similarly, standard deviations, i.e., 0.61 and 0.31, are also low compared with those of other approaches listed in the table. The average dictionary size came out to be 700, reduced from the initial size of 875 atoms, and 2000 atoms, reduced from the initial size of 2500 atoms, respectively. These values are comparable to those of JBDC. Despite the large size of the dictionary for seven samples per class experiment compared with the dictionary size in YaleB, the classification time per test sample, 0.18 ms, is still comparable to that of YaleB. This shows the efficient optimization in the learning of atoms of the Bayesian dictionary through our approach. We did not show the classification time per sample in the case of the experiment with 20 examples per class, as the time will almost proportionally increase with the larger size of the learned dictionary (2000 atoms). We set the sparsity equal to 40 with all other hyperparameters values unchanged, as in Section 9.1. The value of the sparsity increases in this case by the size of the dictionary, which means more atoms are available for the linear combinations for the reconstructions of data and labels.

9.3. Object Classification

Our experiment consists of six stages in which we randomly selected 5, 10, 15, 20, 25, and 30 samples per class, respectively, for training datasets, and the rest in each stage were used as test sets. The results of these experiments are reported in Table 4. The classification accuracy is on the rise with the same trend as observed in the experiments on the YaleB and AR datasets for face recognition. The dictionary size in the experiment stage of 30 samples per class came out to be 3000, which is the same as that of JBDC. However, this value was 3033 for DBDL, and for the best performance of LC-KSVD and D-KSVD, it needs to be 3030 atoms. Just to reveal the effect of our idea of training atoms and the associated parameters in groups, we used this idea in this experiment only due to the comparatively bigger size of the database, Caltech-101, and compared the results with JBDC. It may be noted that we trained the atoms and associated parameters in groups of atoms of size 10 each in inner iterations of Gibbs sampling. Consequently, this reduced the training time by a factor of 65 on average. We visually show the comparative training times in Figure 7

for JBDC and our method. Although we do not show the classification time per sample, we observed that it behaves the same way as in previous experiments. We tested that sparsity in the range of 40 to 70 produces acceptable results. However, lowering the value from 40 degrades the classification accuracy, and increasing the value beyond 70 does not increase the accuracy.

Table 3. Face recognition for AR database [38]. The results are based on two sets of 10 experiments each. In each experiment of the two sets of experiments, 7 and 20 examples were randomly selected for training sets, and the rest of the examples were used as test sets, respectively

Methods/Examples Per Class	**7**		**20**
	Accuracy (%)	**Test Time (ms)**	**Accuracy (%)**
SRC [9]	84.60 ± 1.37	59.91	96.65 ± 1.37
LC-KSVD [8]	85.37 ± 1.34	0.91	96.13 ± 0.64
D-KSVD [23]	85.41 ± 1.49	0.92	96.02 ± 0.58
FDDL [12]	85.97 ± 1.23	50.03	96.22 ± 1.03
ADDL [19]	-	-	96.37 ± 0.78
LC-PDL [20]	-	-	96.38 ± 0.79
SLatDPL [21]	-	-	98.13 ± 0.53
DBDL [25]	86.15 ± 1.19	1.20	97.47 ± 0.99
JBDC [24]	88.90 ± 0.75	0.19	96.70 ± 0.83
Training Time (min)	51.83		
Our Method	**89.27 ± 0.61**	**0.18**	**98.20 ± 0.31**
Training Time (min)	**47.26**		

Table 4. Object classification for the Caltech-101 database [55] with six stages consisting of randomly selected 5, 10, 15, 20, 25, and 30 data points from each class for training sets, respectively.

Training Samples	**5**	**10**	**15**	**20**	**25**	**30**
			Accuracy (%)			
SRC [9]	76.23	79.99	81.27	83.48	84.00	84.51
FDDL [12]	78.31	81.37	83.37	84.76	85.66	85.98
D-KSVD [23]	79.69	83.11	84.99	86.01	86.80	87.72
LC-KSVD [8]	79.74	83.13	85.20	85.98	86.77	87.81
DBDL [25]	80.11	84.03	85.99	86.71	87.97	88.81
JBDC [24]	82.92	89.60	91.65	92.81	93.98	93.82
Training Time (min)	44.13	125.53	222.57	362.38	512	622.44
Our Method	**83.80**	**90.25**	**92.09**	**93.16**	**94.66**	**94.81**
Training Time (min)	**14.28**	**43.02**	**73.67**	**117.61**	**166.96**	**237.05**

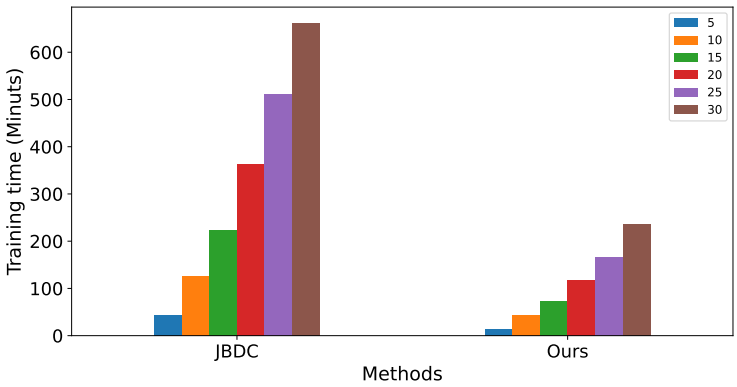

Figure 7. Comparison of the computational cost of JBDC and our approach for the Caltech-101 database

9.4. Scene Categorization

We performed two sets of 10 experiments, each consisting of 50 samples and 20 samples per category, respectively, for training sets and the rest for test sets. We report the results of two sets of 10 experiments each in Table 5. Our approach increases the accuracy by 0.7% in the experiment with 50 samples per class and increases it by 0.3% in the experiment with 20 samples per class when compared with the best accuracy approaches. Although our approach beats LatDPL with a lesser percentage of increase in accuracy, while comparing with the nearest approaches (Bayesian approaches), i.e., DBDL and JBDC, the increase in accuracy is more than 1%. This gain in accuracy is attributed to our idea of learning the same representations for the training data and the corresponding labels over the dictionary and the dictionary classifier, respectively, in the Bayesian setting. Our approach truly exploits the sparse representation and the reconstruction power of a dictionary and tailors a linear classifier to behave like a dictionary classifier. We observed that the learned dictionaries and the dictionary classifiers sizes are 300 and 740 atoms in two experiments for both approaches, i.e., JBDC (the nearest one) and ours, respectively. Despite the same size of the dictionary and the dictionary classifier in both these approaches for 50 samples per class experiment, our approach's test time of 0.66 ms is significantly less than that of JBDC. This clearly shows the efficiency of the learned dictionary of our approach during computing sparse representations of test samples. These representations are ultimately mapped to the corresponding labels as the reconstructions computed by the dictionary classifier. Likewise, the dictionary also performs efficiently in terms of standard deviation, training time, and classification time per test sample. We did not report the test times for 20 samples per class experiments, as the test times follow the trend of the dictionary size, like in the other experiments.

Table 5. Scene categorization for the Fifteen Scene Category database [57]. The results are based on two sets of 10 experiments each. In each experiment of the two sets of experiments, 20 and 50 examples were randomly selected for training sets, and the rest of the examples were used as test sets, respectively.

Methods/Examples	**50**		**20**
Per Class	**Accuracy (%)**	**Test Time (ms)**	**Accuracy (%)**
SRC [9]	95.41 ± 0.13	78.33	
LC-KSVD [8]	95.37 ± 0.28	0.59	93.93 ± 0.45
D-KSVD [23]	95.12 ± 0.18	**0.58**	93.76 ± 0.48
FDDL [12]	94.08 ± 0.43	57.99	—
ADDL [19]	-	-	93.45 ± 0.44
LC-PDL [20]	-	-	93.40 ± 0.45
SLatDPL [21]	-	-	94.63 ± 0.59
DBDL [25]	96.98 ± 0.28	0.71	-
JBDC [24]	97.45 ± 0.28	1.33	93.73 ± 0.41
Training Time (min)	170		37.22
Our Method	**98.10 ± 0.08**	0.66	**94.90 ± 0.54**
Training Time (min)	**160**		**30.11**

9.5. Action Recognition

Our experiment consists of five fivefold 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 and the standard deviation values are reported in Table 6. We reported the results of JBDC and our model. The proposed approach outperformed the other approach in terms of recognition accuracy, standard deviation, training time, and test time.

Table 6. Action recognition for action bank features (processed data) [58] of the UCF sports action database [59] with the five fivefold cross-validations experiment.

Method	Accuracy (%)	Test Time (ms)	Training Time (min)
JBDC [24]	93.43 ± 4.37	15.44	30.32
Our Method	**95.00 ± 1.75**	**4.98**	**22.00**

10. Statistical Analysis

In conjunction with the performance evaluations based on classification accuracies, standard deviations, test times, and training times presented in this paper, here, we present a statistical analysis to obtain deep insight into the method by presenting a comparison with the most relevant method (Bayesian), i.e., JBDC. Although numerous metrics are available to gauge the performance, we present two metrics, i.e., confidence interval and AUC–ROC metrics to evaluate the performance of the methods. Here, AUC stands for the area under the curve, and ROC denotes the receiver operating characteristic curve. We report this analysis in the following sections.

11. Confidence Intervals

We show, in Figure 8, the confidence intervals computed for face recognition experiments on the YaleB and AR databases, and fifteen scene categorization experiments on the Fifteen Scene Category database. We performed 10 experiments in each set of experiments for these databases and computed average classification accuracy along with standard deviation. As the number of experiments is 10 in each set of experiments, and 10 classification accuracy values cannot determine the true mean, we followed the t-test based on a 95% confidence level. Based on the 95% confidence level, we computed the confidence intervals for the methods in each set of experiments. The bar graphs of the means along with confidence intervals bars are shown in Figure 8. We explain the performance of the methods based on confidence intervals in the following sections.

11.1. Confidence Intervals for Yaleb Dataset

Figure 8 presents the mean values along with error bars for the confidence intervals of the methods. The confidence interval for our approach lies at the highest location compared with the locations of the confidence intervals of other methods. Moreover, it does not overlap with any of the other intervals. The width of the confidence interval is also comparable to the intervals of the other methods. Our method predicts with a higher rate than others with a confidence level of 95%. In other words, it outperforms other approaches.

11.2. Confidence Intervals for Ar Database

We performed two sets of experiments for each method. We computed confidence intervals for the set of experiments using 20 examples per class for training for each method. The confidence intervals computed for these methods are visually shown in Figure 8. The confidence interval mean value of our method is higher than the mean values of all other intervals. Though it partially overlaps with one of the intervals, its narrow range and higher position make it prominent for predicting higher true mean values within a small range with a 95% confidence level.

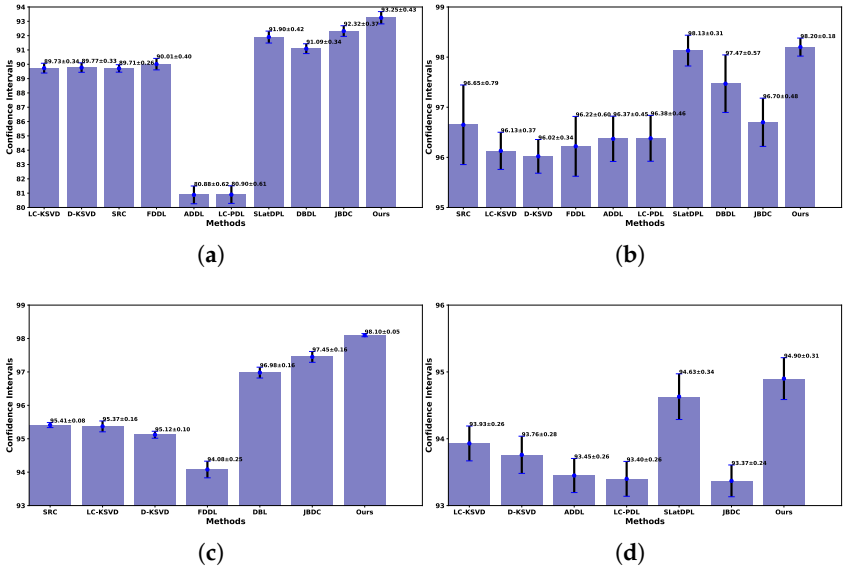

Figure 8. Comparison of Confidence Internals based upon a 95% **confidence level.** (**a**) For face recognition experiment on the Extended YaleB database. (**b**) For face recognition experiment on the AR database. (**c**) For the fifteen scene categorizations experiment on the Fifteen Scene Category database (50 samples/class). (**d**) For the fifteen scene categorizations experiment on the Fifteen Scene Category database (20 samples/class).

11.3. Confidence Intervals for Fifteen Scene Category Database

We performed two sets of experiments for each method. One set of 10 experiments is based upon a random selection of 50 samples per class for training samples. We visually show the confidence intervals of the methods in Figure 8. It clearly shows the highest position of the confidence interval of our method. Moreover, the higher values of the mean lie within a narrow range with a 95% confidence level. Its extraordinary performance is very obvious. We also show the confidence intervals of the set of 10 experiments with 20 examples per class for a training set for each method. The confidence intervals computed for these methods are visually shown in Figure 8. The confidence interval's mean value of our method is higher than the mean values of all other intervals. Though the interval partially overlaps with one of the other intervals, comparatively, its higher position makes it distinctive for predicting higher true mean values with a 95% confidence level.

12. Auc–Roc Analysis

We show, in Figure 9, the ROC (receiver operating characteristic) curves of all the classes based upon One-vs-the-Rest (OvR) settings, considering one class as positive and all others as negative. The curves show the trend of true positive rate with false positive rate for different threshold values for all classes with positive–negative settings. The smaller values of false positive rates and larger values of true positive rates favor the performance of the model for a combination. In other words, a higher AUC (area under the curve) value of the ROC of a class shows the good performance of the classifier in distinguishing between true and false predictions. The average area under the ROC curves of all the positive–negative combinations (all classes) determines the performance of the classification model. We listed the AUC–ROC score of the models in these figures. The figures show that our approach secured scores of 0.992 vs. 0.987 and 0.98 vs. 0.96 for experiments on the Extended YaleB and Fifteen Scene Category databases. The overall score in favor of our approach is also supported by a visual analysis of these figures. In the figures showing the performance of our approach, i.e., Figure 9, the curves bend more towards the left top corner compared with the curves of the JBDC approach. Moreover, the class-wise AUC–ROC scores shown in the bar charts in Figure 10 clearly show the outstanding performance of our approach on

a class-to-class basis. For further insight into the AUC–ROC-based analysis, we provided a glimpse of the numerical figures of a few true positive rates and false positive rates with different threshold values for a few classes in Table 7.

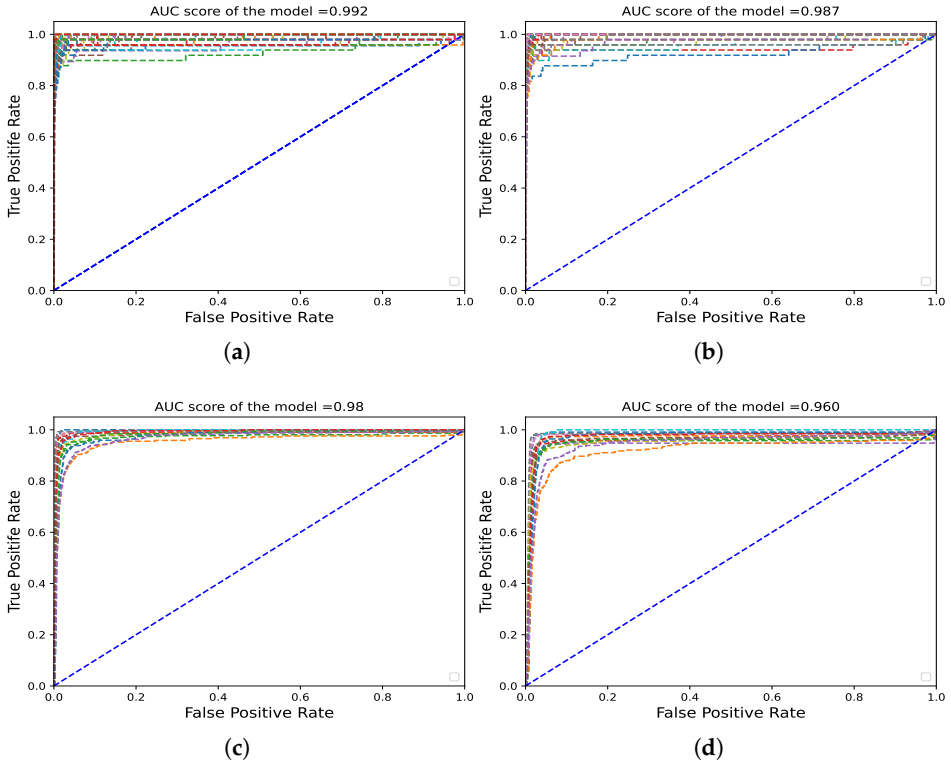

Figure 9. ROC curves based on One-vs-the-Rest (OvR) of the classes. (**a**) Our approach ROC curves for the experiment on the Extended YaleB database. (**b**) JBDC ROC curves for the experiment on the Extended YaleB database. (**c**) Our approach ROC curves for the experiment on the Fifteen Scene Category database. (**d**) JBDC ROC curves for the experiment on the Fifteen Scene Category database.

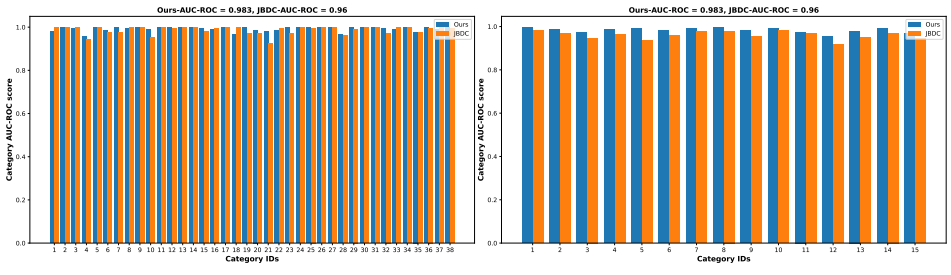

Figure 10. Class-wise AUC–ROC scores for our model and JBDC based on One-vs-the-Rest (OvR) strategy. (**a**) Training on the Face Recognition Extended YaleB database. (**b**) Training on the Fifteen Scene Category database.

Table 7. A glimpse of the AUC/ROC data of four classes (Cat-0, Cat-2, Cat-5, and Cat-12) computed during the training of our model and JBDC on the Fifteen Scene Category database following the One-vs-the-Rest (OvR) setting. These data only reveal a few underlying measures used during the generation of the AUC/ROC data. AUC1 and AUC2 represent the scores for our approach and JBDC, respectively.

Cat.		Ours (Model AUC = 0.983)			JBDC (Model AUC = 0.960)		
		TPR	FPR	Threshold	TPR	FPR	Threshold
cat-0	AUC1 = 0.996, AUC2 = 0.982	0.905	0.007	0.567	0.697	0.009	0.881
		0.905	0.007	0.567	0.719	0.009	0.873
		0.914	0.007	0.553	0.719	0.010	0.872
		0.914	0.007	0.540	0.733	0.010	0.868
		0.928	0.007	0.526	0.733	0.010	0.858
		0.928	0.008	0.520	0.751	0.010	0.837
		0.941	0.008	0.501	0.751	0.010	0.835
		0.941	0.008	0.477	0.778	0.010	0.816
		0.946	0.008	0.471	0.778	0.011	0.813
		0.946	0.008	0.470	0.819	0.011	0.758
		0.950	0.008	0.465	0.819	0.011	0.754
cat-2	AUC1=0.971, AUC2=0.947	0.786	0.006	0.517	0.325	0.007	0.989
		0.805	0.006	0.509	0.325	0.007	0.985
		0.805	0.006	0.509	0.380	0.007	0.932
		0.808	0.006	0.508	0.380	0.007	0.931
		0.808	0.007	0.508	0.403	0.007	0.904
		0.815	0.007	0.496	0.403	0.007	0.903
		0.815	0.007	0.495	0.422	0.007	0.868
		0.818	0.007	0.494	0.422	0.008	0.862
		0.818	0.007	0.490	0.429	0.008	0.860
		0.834	0.007	0.477	0.429	0.008	0.860
		0.834	0.008	0.477	0.442	0.008	0.844
		0.841	0.008	0.475	0.442	0.008	0.839
		0.841	0.008	0.474	0.490	0.008	0.794
cat-5	AUC1=0.983, AUC2=0.959	0.791	0.006	0.587	0.432	0.008	0.974
		0.808	0.006	0.563	0.432	0.008	0.971
		0.808	0.007	0.561	0.500	0.008	0.909
		0.811	0.007	0.561	0.500	0.008	0.909
		0.811	0.007	0.553	0.551	0.008	0.879
		0.839	0.007	0.533	0.551	0.009	0.873
		0.839	0.007	0.530	0.576	0.009	0.856
		0.867	0.007	0.507	0.576	0.009	0.855
		0.867	0.007	0.505	0.579	0.009	0.855
		0.881	0.007	0.495	0.579	0.009	0.853
		0.881	0.008	0.485	0.633	0.009	0.810
Cat-12	AUC1=0.98, AUC2=0.95	0.300	0.005	0.948	0.268	0.007	0.905
		0.300	0.005	0.946	0.268	0.007	0.898
		0.363	0.005	0.887	0.432	0.007	0.800
		0.363	0.005	0.883	0.432	0.008	0.797
		0.400	0.005	0.854	0.468	0.008	0.762
		0.400	0.006	0.853	0.468	0.008	0.761
		0.437	0.006	0.803	0.479	0.008	0.758
		0.437	0.006	0.800	0.479	0.008	0.758
		0.463	0.006	0.782	0.484	0.008	0.753
		0.463	0.006	0.782	0.484	0.008	0.746
		0.542	0.006	0.739	0.511	0.008	0.730

13. Discussion

We trained the same representations for data points and the corresponding labels at the dictionary and the classifier learning stages. Instead of training a new set of representations

for labels, in addition to training representations for training examples at the dictionary learning stage, we further optimize the same representations at the dictionary classifier training stage to tailor them to also represent labels of the corresponding classes. At the prediction stage, the representations of test samples are computed over the dictionary, and the same representations are used as input to the dictionary classifier for classifications. In the case of two sets of representations, the second set of representations learned at the dictionary classifier level to represent the labels is not available at the prediction stage. In the case of using the same representations learned at the dictionary stage that have also been optimized for classification, this will enhance the classification accuracy. The results confirm our claim of improvement in accuracy. Additionally, our model also gives the additive advantage of the improvement in training time, in conjunction with the improvement in classification accuracy. Our approach frees computational resources that would have been engaged in learning the second set of representations at the classifier level, resulting in saving training time.

Moreover, we also tested our idea of learning dictionary and classifier atoms and associated parameters in groups for the Caltech-101 database only, as this database is comparatively bigger. We only compared the outcomes of this idea with JBDC just to reveal its effect. Following this idea, we reduced the training time by a factor of 65 on average.

We tuned the hyperparameters of our model in conjunction with the theoretical background mentioned in [24]. Accordingly, we conclude $0 < a_0, b_0 < \min_c |I_c|$. Ideally, we need $N \to \infty$, but $K > N$ is sufficiently large enough to serve the same purpose for initialization of the size of the dictionary and the classifier. Its value finally reduces to a number of dictionary atoms fewer than N, as the result of dictionary atoms pruning during iterations of the Gibbs sampler. Analytical proof of dictionary atoms pruning is given by [24]. We placed noninformative gamma priors on λ_a and λ_h. These parameters of the model are adaptively learned to the values that produce the best classification accuracies. The initialization of these parameters within a range of 10^6–10^9 produces acceptable results. However, values below 10^6 degrade the classification accuracy, but higher values are acceptable. We found that the value 10^9 produces the best results, except for the UCF sports action database, in which case the value 10^{12} produces the best results. These values explored in the learning process suit our data because most of the data are clean. The higher value in the case of the UCF sports action database is also because of the availability of only small amounts of training data. It may be noted that our data also carry noise due to variance in illuminations and other factors. Our model also takes into consideration these real-life problems and learns the model parameters adaptively to produce the best results. We also placed noninformative gamma prior on λ_s^c, and this parameter is learned accordingly. We observed that the initialization of λ_s^c with a value around 1 produces the best sparse codes. We set the value of the precision hyperparameter λ_{ϕ_0} for the dictionary equal to the dimension of the atoms, i.e., M. Similarly, we set the value of the precision hyperparameter λ_{b_0} equal to C. Small deviations in these values do not affect the results. However, large deviations on the order of 20 degrade the results. We observed that sparsity increases with increases in the sizes of the learned dictionaries. However, the values smaller than the threshold values degrade the results. The threshold values that we found are 30, 40, 50, 40, and 30 for the YaleB, AR face, Caltech-101, Fifteen Scene, and UCF sports action datasets. We set the values of the noninformative hyperparameters c_0, d_0, e_0, and f_0 equal to 10^6. No considerable change in classification accuracies was found by deviating the values.

14. Conclusions

To solve our Bayesian network for posterior parameters, we used the Gibbs sampler as an inference technique. The results are found to be improved compared with other state-of-the-art approaches. This improvement in the results is attributed to the idea we presented for learning the same weights for the representation of data samples and the corresponding labels. We also achieved a gain in training time as an added advantage,

as we reduced the computational cost by learning one set of weights for the representation of data and labels at the dictionary and classifier learning stages. It was observed that the formulation of the problem in Bayesian settings provides us an advantage of using the robust optimization algorithm, i.e., the Gibbs sampler for efficiently solving probabilistic Bayesian networks.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app14010306/s1.

Author Contributions: Conceptualization, M.R.-u.-d. and F.S.; Methodology, M.R.-u.-d.; Software, M.R.-u.-d.; Formal analysis, S.A.G.; Investigation, M.R.-u.-d.; Data curation, M.R.-u.-d.; Writing—original draft, M.R.-u.-d.; Writing—review & editing, S.A.G.; Supervision, F.S.; Project administration, F.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Supplementary material is attached to this manuscript. Code and data are available at https://doi.org/10.5281/zenodo.10059916 (accessed on 1 November 2023).

Acknowledgments: We thank all the researchers at the "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.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Song, P.; Rodrigues, M.R. Multimodal Image Denoising Based on Coupled Dictionary Learning. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 515–519. [CrossRef]
2. Li, J.; Wang, J.; Li, J. Image Denoising Algorithm Based on Incoherent Dictionary Learning. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 3337–3340. [CrossRef]
3. Elad, M.; Aharon, M. Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries. *IEEE Trans. Image Process.* **2006**, *15*, 3736–3745. [CrossRef] [PubMed]
4. Mairal, J.; Elad, M.; Sapiro, G. Sparse Representation for Color Image Restoration. *IEEE Trans. Image Process.* **2008**, *17*, 53–69. [CrossRef] [PubMed]
5. Lin, F.; Fei, Z.; Wan, J.; Wang, N.; Chen, D. A Robust Efficient Dictionary Learning Algorithm for Compressive Data Gathering in Wireless Sensor Networks. In Proceedings of the 2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 26–27 August 2017; Volume 2, pp. 12–15. [CrossRef]
6. Xu, K.; Li, Y.; Ren, F. An energy-efficient compressive sensing framework incorporating online dictionary learning for long-term wireless health monitoring. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 804–808. [CrossRef]
7. Ge, J.; Zhou, T.; Zhang, F.; Tse, K. Learning Part-Based Dictionary by Sparse NMF for Face Gender Recognition. In Proceedings of the 2015 8th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 12–13 December 2015; Volume 2, pp. 375–378. [CrossRef]
8. Jiang, Z.; Lin, Z.; Davis, L.S. Label Consistent K-SVD: Learning a Discriminative Dictionary for Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2651–2664. [CrossRef] [PubMed]
9. Wright, J.; Yang, A.Y.; Ganesh, A.; Sastry, S.S.; Ma, Y. Robust Face Recognition via Sparse Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 210–227. [CrossRef] [PubMed]
10. Moeini, H.; Mozaffari, S. Gender dictionary learning for gender classification. *J. Vis. Commun. Image Represent.* **2017**, *42*, 1–13. [CrossRef]
11. Jian, Z.; Chao, Z.; Shunli, Z.; Tingting, L.; Weiwen, S.; Jian, J. Pre-detection and dual-dictionary sparse representation based face recognition algorithm in non-sufficient training samples. *J. Syst. Eng. Electron.* **2018**, *29*, 196–202. [CrossRef]
12. Yang, M.; Zhang, L.; Feng, X.; Zhang, D. Sparse Representation Based Fisher Discrimination Dictionary Learning for Image Classification. *Int. J. Comput. Vis.* **2014**, *109*, 209–232. [CrossRef]
13. Yang, M.; Zhang, L.; Yang, J.; Zhang, D. Robust sparse coding for face recognition. In Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; IEEE Computer Society: Piscataway, NJ, USA, 2011; pp. 625–632. [CrossRef]

14. Yang, M.; Zhang, L.; Yang, J.; Zhang, D. Metaface learning for sparse representation based face recognition. In Proceedings of the International Conference on Image Processing, ICIP 2010, Hong Kong, China, 26–29 September 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 1601–1604. [CrossRef]
15. Zhang, J.; Shum, H.P.H.; Han, J.; Shao, L. Action Recognition from Arbitrary Views Using Transferable Dictionary Learning. *IEEE Trans. Image Process.* **2018**, *27*, 4709–4723. [CrossRef]
16. Castrodad, A.; Sapiro, G. Sparse Modeling of Human Actions from Motion Imagery. *Int. J. Comput. Vis.* **2012**, *100*, 1–15. [CrossRef]
17. Wilson, S.; Mohan, C.K. Coherent and Noncoherent Dictionaries for Action Recognition. *IEEE Signal Process. Lett.* **2017**, *24*, 698–702. [CrossRef]
18. Wang, H.; Yuan, C.; Hu, W.; Sun, C. Supervised class-specific dictionary learning for sparse modeling in action recognition. *Pattern Recognit.* **2012**, *45*, 3902–3911. [CrossRef]
19. Zhang, Z.; Jiang, W.; Qin, J.; Zhang, L.; Li, F.; Zhang, M.; Yan, S. Jointly Learning Structured Analysis Discriminative Dictionary and Analysis Multiclass Classifier. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 3798–3814. [CrossRef] [PubMed]
20. Zhang, Z.; Jiang, W.; Zhang, Z.; Li, S.; Liu, G.; Qin, J. Scalable Block-Diagonal Locality-Constrained Projective Dictionary Learning. *arXiv* **2019**, arXiv:1905.10568.
21. Zhang, Z.; Sun, Y.; Wang, Y.; Zhang, Z.; Zhang, H.; Liu, G.; Wang, M. Twin-Incoherent Self-Expressive Locality-Adaptive Latent Dictionary Pair Learning for Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 947–961. [CrossRef] [PubMed]
22. Wang, D.; Kong, S. A classification-oriented dictionary learning model: Explicitly learning the particularity and commonality across categories. *Pattern Recognit.* **2014**, *47*, 885–898. [CrossRef]
23. Zhang, Q.; Li, B. Discriminative K-SVD for dictionary learning in face recognition. In Proceedings of the The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13–18 June 2010; IEEE Computer Society: Piscataway, NJ, USA, 2010; pp. 2691–2698. [CrossRef]
24. Akhtar, N.; Mian, A.; Porikli, F. Joint Discriminative Bayesian Dictionary and Classifier Learning. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3919–3928. [CrossRef]
25. Akhtar, N.; Shafait, F.; Mian, A. Discriminative Bayesian Dictionary Learning for Classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2374–2388. [CrossRef] [PubMed]
26. Li, W.; Liang, J.; Wu, Q.; Zhou, Y.; Xu, X.; Wang, N.; Zhou, Q. An efficient face classification method based on shared and class-specific dictionary learning. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 2596–2600. [CrossRef]
27. Zhou, N.; Shen, Y.; Peng, J.; Fan, J. Learning inter-related visual dictionary for object recognition. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3490–3497. [CrossRef]
28. Yan, L.; Zhu, R.; Liu, Y.; Mo, N. Class-Specific Dictionary Based Semi-Supervised Domain Adaptation for Land-Cover Classification of Aerial Images. In Proceedings of the IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 720–723. [CrossRef]
29. Pan, F.; Zhang, Z.X.; Liu, B.D.; Xie, J.J. Class-Specific Sparse Principal Component Analysis for Visual Classification. *IEEE Access* **2020**, *8*, 110033–110047. [CrossRef]
30. Mairal, J.; Bach, F.R.; Ponce, J.; Sapiro, G.; Zisserman, A. Discriminative learned dictionaries for local image analysis. In Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), Anchorage, AK, USA, 24–26 June 2008; IEEE Computer Society: Piscataway, NJ, USA, 2008. [CrossRef]
31. Qiu, Q.; Jiang, Z.; Chellappa, R. Sparse dictionary-based representation and recognition of action attributes. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 707–714. [CrossRef]
32. Mairal, J.; Bach, F.; Ponce, J.; Sapiro, G.; Zisserman, A. Supervised Dictionary Learning. *Adv. Neural Inf. Process. Syst.* **2008**, *21*.
33. Liu, H.; Liu, H.; Sun, F.; Fang, B. Kernel Regularized Nonlinear Dictionary Learning for Sparse Coding. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 766–775. [CrossRef]
34. Jenatton, R.; Mairal, J.; Obozinski, G.; Bach, F. Proximal Methods for Hierarchical Sparse Coding. *J. Mach. Learn. Res.* **2011**, *12*, 2297–2334.
35. Liu, H.; Sun, F.; Guo, D.; Fang, B.; Peng, Z. Structured Output-Associated Dictionary Learning for Haptic Understanding. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 1564–1574. [CrossRef]
36. Deng, W.; Hu, J.; Guo, J. Face Recognition via Collaborative Representation: Its Discriminant Nature and Superposed Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2513–2521. [CrossRef] [PubMed]
37. Ramirez, I.; Sprechmann, P.; Sapiro, G. Classification and clustering via dictionary learning with structured incoherence and shared features. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 3501–3508. [CrossRef]
38. Martinez, A.; Benavente, R. *The RA Face Database*; CVC Technical Report 24; Universitat Autònoma de Barcelona: Bellaterra, Spain, 1998.
39. Mairal, J.; Bach, F.; Ponce, J. Task-Driven Dictionary Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 791–804. [CrossRef]
40. Yang, J.; Yu, K.; Huang, T.S. Supervised translation-invariant sparse coding. In Proceedings of the The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13–18 June 2010; IEEE Computer Society: Piscataway, NJ, USA, 2010; pp. 3517–3524. [CrossRef]

41. Jiang, Z.; Lin, Z.; Davis, L.S. Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 1697–1704. [CrossRef]
42. Zhang, Z.; Xu, Y.; Shao, L.; Yang, J. Discriminative Block-Diagonal Representation Learning for Image Recognition. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 3111–3125. [CrossRef] [PubMed]
43. Zhou, M.; Chen, H.; Paisley, J.; Ren, L.; Li, L.; Xing, Z.; Dunson, D.; Sapiro, G.; Carin, L. Nonparametric Bayesian Dictionary Learning for Analysis of Noisy and Incomplete Images. *IEEE Trans. Image Process.* **2012**, *21*, 130–144. [CrossRef] [PubMed]
44. Akhtar, N.; Mian, A. Nonparametric Coupled Bayesian Dictionary and Classifier Learning for Hyperspectral Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 4038–4050. [CrossRef] [PubMed]
45. Yang, L.; Fang, J.; Cheng, H.; Li, H. Sparse Bayesian Dictionary Learning with a Gaussian Hierarchical Model. *Signal Process.* **2015**, *130*, 93–104. [CrossRef]
46. Wen, S.; Liu, W.; Yang, Y.; Zhou, P.; Guo, Z.; Yan, Z.; Chen, Y.; Huang, T. Multilabel Image Classification via Feature/Label Co-Projection. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 7250–7259. [CrossRef]
47. Wang, X.; Bao, A.; Lv, E.; Cheng, Y. Multiscale Multipath Ensemble Convolutional Neural Network. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 5918–5928. [CrossRef]
48. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
49. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
50. Pham, D.; Venkatesh, S. Joint learning and dictionary construction for pattern recognition. In Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), Anchorage, AK, USA, 24–26 June 2008; IEEE Computer Society: Piscataway, NJ, USA, 2008. [CrossRef]
51. Paisley, J.W.; Carin, L. Nonparametric factor analysis with beta process priors. In Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, QC, Canada, 14–18 June 2009; Danyluk, A.P., Bottou, L., Littman, M.L., Eds.; ACM International Conference Proceeding Series; ACM: New York, NY, USA, 2009; Volume 382, pp. 777–784. [CrossRef]
52. Zhou, M.; Chen, H.; Ren, L.; Sapiro, G.; Carin, L.; Paisley, J.W. Non-Parametric Bayesian Dictionary Learning for Sparse Image Representations. In *Advances in Neural Information Processing Systems 22*; Bengio, Y., Schuurmans, D., Lafferty, J.D., Williams, C.K.I., Culotta, A., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2009; pp. 2295–2303.
53. Pati, Y.C.; Rezaiifar, R.; Krishnaprasad, P.S. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 1–3 November 1993; pp. 40–44. [CrossRef]
54. Georghiades, A.S.; Belhumeur, P.N.; Kriegman, D.J. From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 643–660. [CrossRef]
55. Li, F.; Fergus, R.; Perona, P. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Comput. Vis. Image Underst.* **2007**, *106*, 59–70. [CrossRef]
56. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
57. Lazebnik, S.; Schmid, C.; Ponce, J. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), New York, NY, USA, 17–22 June 2006; IEEE Computer Society: Piscataway, NJ, USA, 2006; pp. 2169–2178. [CrossRef]
58. Sadanand, S.; Corso, J.J. Action Bank: A High-Level Representation of Activity in Video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012.
59. Rodriguez, M.D.; Ahmed, J.; Shah, M. Action MACH a spatio-temporal Maximum Average Correlation Height filter for action recognition. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8. [CrossRef]
60. Cai, T.T.; Wang, L. Orthogonal Matching Pursuit for Sparse Signal Recovery with Noise. *IEEE Trans. Inf. Theory* **2011**, *57*, 4680–4688. [CrossRef]