A META-ALGORITHM FOR CLASSIFICATION BY FEATURE NOMINATION

Rituparna Sarkar, Kevin Skadron and Scott T. Acton

Electrical and Computer Engineering, University of Virginia
Computer Science Department, University of Virginia
Charlottesville, VA, USA

ABSTRACT

With increasing complexity of the dataset it becomes impractical to use a single feature to characterize all constituent images. In this paper we describe a method that will automatically select the appropriate image features that are relevant and efficacious for classification, without requiring modifications to the feature extracting methods or the classification algorithm. We first describe a method for designing class distinctive dictionaries using a dictionary learning technique, which yields class specific sparse codes and a linear classifier parameter. Then, we apply information theoretic measures to obtain the more informative feature relevant to a test image and use only that feature to obtain final classification results. With at least one of the features classifying the query accurately, our algorithm chooses the correct feature in 88.9% of the trials.

***Index Terms*—dictionary learning, classification, sparse representation, conditional entropy, feature nomination.**

1. INTRODUCTION

Standard image retrieval or classification techniques generally follow a two-step approach. First, a set of discriminative feature descriptors is chosen to efficiently represent the objects in the test image, and then, the selected features are input to a classifier, which determines the class or label of the test image. Efficacy of these systems relies on accurate and discriminative feature selection, as well as proper design of the classifier.

However, for complicated datasets, the task of selecting one representative feature vector is often non-trivial. Complexity of a dataset refers to variability in contents of the images belonging to the same class and also between images of different classes. As an example, a dataset may have flags of countries as well as buildings. While color features can differentiate flags of countries, buildings may need local descriptors to capture the structural differences. Depending on the complexity of the database items, it may be almost impossible to correctly represent an item based on a single feature selection technique. This calls for feature boosting strategies, where multiple feature selection routines are combined to generate the feature vector set. An approach to solve for the intra-class scatter of image properties is to select the optimal set of features discriminative of a class. Such feature selection methods for enhancing image retrieval performance via retaining only the more informative features for a class via maximizing mutual information have been discussed in [1] [2] [3] [4] [5]. In [6] a method of hierarchically arranging image features according to relevance for a particular class is discusses. One common aspect of these methods is that the algorithms emphasize the selection of the optimal set of features from all the images by one particular feature selection technique. These strategies suffer from a particular drawback which renders the above mentioned methods unreliable for classification and retrieval purposes for databases characterized by significant content variability. This is chiefly because one particular set of feature descriptor may not be sufficiently discriminative for all the categories of objects present in the database.

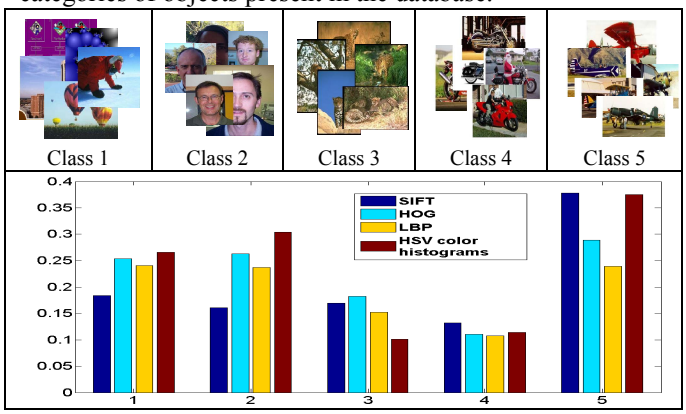


Fig. 1: The first row denotes 5 classes from Caltech101 dataset. The 3rd row shows precision results shown for these 5 classes using SIFT **[7]**, HOG **[8]**, LBP **[9]** and color histograms. The precision results obtained are average precision of all the images in a class. (*The graph is best viewed in color*).

As shown in Fig.1 for different classes, classification accuracy changes with the feature type. With greater intra-class complexity, features extracted by one particular method may not be discriminative enough to represent one class, in case of which images belonging to same class can

be discriminated by different feature types. Motivated by this fact, we design a system, which is capable of choosing the appropriate feature given a test image for accurate classification based on sparse representation. Exploiting sparse codes for classification purposes has been discussed in [10], where the test sample is represented as a linear combination of training samples. Furthermore in [11] [12] [13], it has been shown that a discriminative dictionary learned from the images can be used for sparse representation and classification purpose.

In this paper, we discuss a method for designing compact and class-specific dictionary that can be utilized for classification. The original features can then be represented as a linear combination of this dictionary where the features from the same class share a common dictionary atom making it more class distinctive. Simultaneously, from this dictionary learning algorithm, we obtain a classifier weight matrix for classifying the test image. A relevance measure between features and the class to which they belong can be obtained by maximizing mutual information. So, finally for a given test image, once the sparse codes for different features and corresponding class labels are determined, we deploy an information theoretic technique for selecting the most relevant feature.

2. DISCRIMINATIVE FEATURE SELECTION

Sparse representation based dictionary learning has gained popularity in the recent years. Sparse coding can be efficiently utilized by representing a feature vector Y as a linear combination of some basis vectors. This can be written as $Y = DX$, where D is a matrix in which columns represent the basis vectors, and X contains the representative sparse codes.

Let us define a matrix $Y = [Y_1, Y_2, \dots\dots, Y_C]$, where C is the number of classes present in the dataset. Here $Y_i = [\boldsymbol{y}_1, \boldsymbol{y}_2, \dots\dots, \boldsymbol{y}_{N_i}]$, $(Y_i \in \mathbb{R}^{n \times N_i})$. $\boldsymbol{y}_v \in \mathbb{R}^n$ denotes a feature vector for an v^{th} image in i^{th} class containing N_i images, *i.e.*, $v = 1 \dots . N_i$.

The columns of a dictionary D serve as the basis vectors for representing$Y$ and can be exploited to obtain the sparse code for the test images. D can be learned from the set of training examples [11] [12] [13] [14]. The dictionary can be written as $D = [D_1, D_2, \dots\dots, D_C]$, $D_i \in \mathbb{R}^{n \times K}$ is the sub-dictionary representative of each class.

Let $\boldsymbol{x}_v \in \mathbb{R}^M$ $(M = KC)$ be the sparse code for representing $\boldsymbol{y}_v$. The sparse codes for a class can be embedded in the matrix $X_i = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots\dots, \boldsymbol{x}_{N_i}]$, $X_i \in \mathbb{R}^{M \times N_i}$. $X = [X_1, X_2, \dots\dots, X_C]$, denote the sparse codes for the dataset.

Sparse representation based dictionary learning [14] is accomplished by learning a dictionary D and obtaining a sparse code X for a given input data Y by minimizing the following

$$\underset{X,D}{\operatorname{argmin}} \|Y - DX\|_2^2 \;\; \text{s.t} \;\; \|\boldsymbol{x}_v\|_0 \leq t \;\forall v \tag{1}$$

Here t is the upper bound on the number of non-zero elements of the sparse vector $\boldsymbol{x}_v$.

2.1. Discriminative dictionary learning and classification

The dictionary learning method featuring the K-SVD [14] algorithm, as in (1), minimizes the reconstruction error with a sparsity constraint on $\boldsymbol{x}_v$ given a signal $\boldsymbol{y}_v$. However, (1) does not include any constraint that can discriminate between two different signals making it unsuitable for classification or image retrieval purposes. This necessitates a specialized technique for dictionary learning. We introduce a dictionary learning scheme, which can be utilized for classification purpose. The purpose is to build class representative dictionary, so that sparse codes generate for features belonging to the same class, using this dictionary, share similar dictionary atoms. We solve the following optimization to obtain the desired dictionary.

$$\underset{X,D,A,W}{\operatorname{argmin}} \mathcal{C}(X, D, A, W)$$

$$\begin{aligned} \mathcal{C}(X, D, A, W) = \quad & \|Y - DX\|_2^2 + \gamma \left\|\dot{\boldsymbol{\mathcal{X}}} - \boldsymbol{I}X\right\|_2^2 + \\ & \alpha\|Q - AX\|_2^2 + \beta\|H - WX\|_2^2 \end{aligned} \tag{2}$$

$$\text{s.t} \;\; \|x_v\|_0 \leq t \;\; \forall v$$

Here $\dot{\boldsymbol{\mathcal{X}}}$ ensures that the sparse codes x_v are bounded along each dimension. This reduces the disparity between the sparse codes of training and test data. Along with sharing the same dictionary atoms, it minimizes the error of the entry along each dimension of the sparse codes of same class. The bound is determined by the sparse codes obtained solving the following

$$\underset{\boldsymbol{\mathcal{X}}_i, D_i}{\operatorname{argmin}} \|Y_i - D_i\boldsymbol{\mathcal{X}}_i\|_2^2 \; s.t \; \forall k = \{1 \dots N_i\}, \|x_k\|_0 \leq t \tag{3}$$

Here $\boldsymbol{\mathcal{X}}_i \in \mathbb{R}^{K \times N_i}$ is the sparse code generated for class i. Then, $\dot{\boldsymbol{\mathcal{X}}} = \begin{bmatrix} \boldsymbol{\mathcal{X}}_1 & \dots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \dots & \boldsymbol{\mathcal{X}}_C \end{bmatrix}, \boldsymbol{0} \in \mathbb{R}^{K \times N_i}$. $\boldsymbol{I} \in \mathbb{R}^{M \times M}$, is an identity matrix. $Q = [Q_1, Q_2, \dots\dots, Q_C]$, $Q_i \in \mathbb{R}^{KC \times N_i}$,as in [13], is the label determining the pair of dictionary atom and signal sharing the same class. $Q_i(a, b) = 1$ if $\boldsymbol{d}_a$ and $\boldsymbol{y}_b$ are the dictionary atom and training data represents class i. A is a transformation matrix that would regularize the sparse codes of the same class to share similar dictionary atoms. H is the matrix containing the class labels i.e., $H(i, b) = 1$ if $\boldsymbol{y}_b$ is a member of class i [12] [13]. Here we assume a linear classifier model; the label of an input signal is given as:

$$(\ell(\boldsymbol{y}_v) = i) = \underset{i}{\operatorname{argmax}}(W^T\boldsymbol{x}_v) \tag{4}$$

W is the classifier determinant parameter, which regularizes the sparse codes from same class to share similar dictionary atoms. A and W are initialized [13] [12] as shown in the following equation:

$$\begin{aligned} A &= \left(\dot{\boldsymbol{\mathcal{X}}}\dot{\boldsymbol{\mathcal{X}}}^T + \lambda_1\boldsymbol{I}\right)^{-1}\dot{\boldsymbol{\mathcal{X}}}Q^T \\ W &= \left(\dot{\boldsymbol{\mathcal{X}}}\dot{\boldsymbol{\mathcal{X}}}^T + \lambda_2\boldsymbol{I}\right)^{-1}\dot{\boldsymbol{\mathcal{X}}}H \end{aligned} \tag{5}$$

In Fig. 2, we show classification accuracy (ratio of number of correct classification to the total number of test images) using the method described here for four different feature descriptors. Once the classification results are obtained for the four features, our next goal is to nominate the feature that has classified the test image accurately.

2.2. Selecting feature descriptor

It can be seen from Fig. 2: Classification accuracy for four sample classes of Caltech 101 dataset using (2) for different features before feature nomination. A comparison with LC-KSVD2 is given in the rightmost column.that for different classes of images, accuracy for classification is dependent on the choice of feature descriptor. This necessitates that the appropriate feature descriptor be chosen for a given query type to reduce the chances of undesired classification. We propose an information theoretic approach to dynamically choose the feature descriptor based on a given query type and the image contents.

As mentioned earlier a relevance measure between features and the class they belong can be obtained by maximizing the mutual information [1],[2], [3], [4], [5]. For a given feature $\boldsymbol{x}$ the mutual information between the feature and its class $\ell(\boldsymbol{x}) = i$ is given by (4).

$$\mathbb{I}(\boldsymbol{x}, \ell(\boldsymbol{x}) = i) = H(i) - H(i|\boldsymbol{x}) \tag{4}$$

where $H(i)$ is the entropy given by:

$$H(\boldsymbol{x}) = p(\boldsymbol{x}) \log\left(\frac{1}{p(\boldsymbol{x})}\right) \tag{5}$$

For any class i the class probability is given as, $p(i) = \frac{N_i}{N}, i = \{1 \dots . C\}$. We keep the the number of training features per class constant which implies that the entropy of a class is also constant. Thus maximizing the mutual information between a feature and a class would mean minimizing the conditional entropy $H(i|\boldsymbol{x})$. The conditional entropy is given by:

$$H(i|\boldsymbol{x}) = p(i|\boldsymbol{x}) \log\left(\frac{1}{p(i|\boldsymbol{x})}\right) = \frac{p(\boldsymbol{x}|i)p(i)}{p(\boldsymbol{x})} \log\left(\frac{p(\boldsymbol{x})}{p(\boldsymbol{x}|i)p(i)}\right) \tag{6}$$

The class conditional probability measure for a feature can be estimated by using a Parzen window technique [15] using a Gaussian kernel as shown in (7).

$$p(\boldsymbol{x}|i) = \frac{1}{N_i} \sum_{v=1}^{N_i} \mathcal{K}(\boldsymbol{x} - x_v, \Sigma) \tag{7}$$

Where, $\mathcal{K}(\boldsymbol{x} - x_v, \Sigma) = \frac{1}{(2\pi)^{\frac{M}{2}} |\Sigma|^{\frac{1}{2}}} e^{-(x-x_v)^T \Sigma^{-1} (x-x_v)}$

x_v refers to a member of the training data of class i and the marginal is given as $p(\boldsymbol{x}) = \sum_{i=1}^{C} p(\boldsymbol{x}|i)p(i)$.

When a feature descriptor for the test data $\boldsymbol{x}$ and its class label i is available, the mutual information provides a measure of certainty of $\boldsymbol{x}$ belonging to class i.

3. FEATURE NOMINATION

3.1. Classification and feature extraction

A single feature, in most of the cases, cannot classify images in a given class accurately. Hence, to adequately classify an object, the appropriate feature must be chosen.

We define a feature descriptor type F_l where $l = 1 \dots \dots . L$ and L denotes the number of feature types being used for classification. For our experiments we use four features F_1: SIFT [7], F_2: Histogram of oriented gradients (HOG) [8], F_3: local binary pattern (LBP) [9], and F_4: HSV color histograms. We use our feature nomination algorithm to choose between these four features to provide the ultimate classification result.

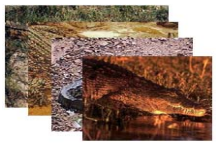
Image classes	New meta-algorithm for classification		LC-KSVD2 [13]	
	SIFT HOG LBP HSV color hist.	100% 80% 92% 8%	SIFT HOG LBP HSV color hist	100% 80% 84% 8%
	SIFT HOG LBP HSV color hist	100% 91% 93% 0%	SIFT HOG LBP HSV color hist	100% 92% 91% 0%
	SIFT HOG LBP HSV color hist	100% 100% 77% 5%	SIFT HOG LBP HSV color hist	100% 100% 75% 5%
	SIFT HOG LBP HSV color hist	82% 53% 41% 14%	SIFT HOG LBP HSV color hist	76% 52% 29% 14%

Fig. 2: Classification accuracy for four sample classes of Caltech 101 dataset using (2) for different features before feature nomination. A comparison with LC-KSVD2 **[13]** is given in the rightmost column.

The feature vector $Y^l = [Y_1^l, Y_2^l, \dots \dots, Y_C^l]$ corresponds to feature type l, for classes $1 \dots . C$. The respective sparse codes are $X^l = [X_1^l, X_2^l, \dots \dots, X_C^l]$. The sparse codes for a particular feature descriptor l is obtained by solving

$$\underset{X^l, D^l, A^l, W^l}{\operatorname{argmin}} \; \mathcal{C}(X^l, D^l, A^l, W^l)$$

$$\mathcal{C}(X^l, D^l, A^l, W^l) = \|Y^l - D^l X^l\|_2^2 + \gamma \left\| \dot{\boldsymbol{X}}^l - \boldsymbol{I} X^l \right\|_2^2 + \alpha \|Q - A^l X^l\|_2^2 + \beta \|H - W^l X^l\|_2^2 \tag{8}$$

As the number of features in the training set remains the same irrespective of the feature descriptor type, Q, H which correlate between the features and their classes, remain same. For a given query image q, the feature descriptor y_q^l for feature type l is computed and the respective sparse code $\boldsymbol{x}_q^l$ is obtained by solving,

$$\underset{x_q^l}{\operatorname{argmin}} \left\| \boldsymbol{y}_q^l - D^l \boldsymbol{x}_q^l \right\|_2^2 \text{ s.t } \left\| \boldsymbol{x}_q^l \right\|_0 \leq t \tag{9}$$

The feature specific class label for the test image is given by

$$(\ell(x_q^l) = i) = \max_i((W^l)^T \boldsymbol{x}_q^l) \quad (10)$$

3.2. Feature nomination

Once the class labels corresponding to the feature descriptors F_l are obtained, it is required to identify the most relevant class for the query. Comparing the class conditional densities, a measure of how likely the test image will actually belong to the class label assigned to it, can be obtained. The class conditional entropy can either be computed by the original feature or the sparse codes obtained by solving (9). To account for the any loss of information that may have incurred due to sparse coding of x_q^l, we compare $H\left(y_q^l \middle| \ell(x_q^l)\right) H\left(x_q^l \middle| \ell(x_q^l)\right)$ for all l. Thus the final classification result is given by the nominated feature type l :

$$\ell(q) = \min_l H\left(y_q^l \middle| \ell(x_q^l)\right) H\left(x_q^l \middle| \ell(x_q^l)\right) \quad (11)$$

4. EXPERIMENTAL RESULTS

Experiments were performed using the Caltech101 dataset, which contains (Fei-Fei, Fergus and Perona) 101 different categories with 9,144 images. The number of images in a class varies from 31 to 800. We choose randomly selected 28 images per class to train the classifier for each of SIFT, HOG, LBP and HSV color histograms. The remaining images were used as test images. For SIFT we extract the features in similar lines with (Jiang, Lin and Davis). We first compute the SIFT features on 16x16 grid with spacing of 2 pixels. Then we compute the spatial pyramid (Lazebnik, Schmid and Ponce) structure for 3 levels, breaking the image into 4 blocks and then into 8 blocks. Then, the dimensionality of the extracted features was finally reduced using PCA.

	Faces	Leopards	Accordion	Cellphone	Dollar bill	Euphonium	Garfield	Joshua tree	Laptop	Metronome	Minaret	Okapi	Pagoda	Rooster	Scissors	Sunflower
Faces	0.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01
Leopards	0	0.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Accordion	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Cellphone	0	0	0	0.9	0	0	0	0	0	0	0	0	0	0	0	0
Dollar bill	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Euphonium	0	0	0	0	0	0.8	0	0	0	0	0	0	0	0	0	0
Garfield	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Joshua tree	0	0	0	0	0	0	0	0.9	0	0	0	0	0	0	0	0
Laptop	0	0	0	0	0	0	0	0	0.9	0	0	0	0	0	0	0
Metronome	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Minaret	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Okapi	0	0	0	0	0	0	0	0	0	0	0	0.9	0	0	0	0
Pagoda	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Rooster	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Scissors	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Sunflower	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.85

Fig. 3: The figure shows the confusion matrix (the diagonal entries show the classification accuracy when a test image from the classes along the row is classified correctly) for 16 sample classes which have classification accuracy over 80% using the feature the feature nomination scheme.

For HOG features, we compute the spatial pyramid by concatenating the histograms of the first, second and third level i.e., by breaking the image in 1x1, 3x3 and 5x5 blocks. Similar features were computed using LBP and color histograms, but only two levels were used to create the spatial pyramid structure. The sparse codes and the class labels we obtained using these four features. Finally the feature descriptor voting using the conditional entropy was accomplished using these sparse codes and the features for the obtained class labels.

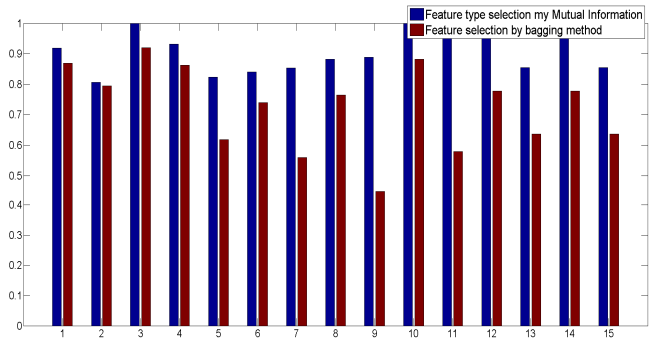


Fig. 4: Comparison of classification accuracy (number of correct class predictions/number of test images in that class) between our feature selection scheme and bagging algorithm is shown for 10 sample classes

In Fig. 3, we show accuracy percentage using feature descriptor voting scheme for 16 sample classes which have accuracy more that 80%. About 10% of the classes for the dataset have 100% accuracy and 12.7% classes have more than 90% accuracy. Assuming that accurate class labels will be obtained for at the least one of the feature descriptor type, out feature voting scheme chooses the correct class for 88.93% cases. A comparison using the bagging predictor [18] with our classification algorithm is shown in Fig. 4. In our case, once the class label for each feature is obtained using the predictor, the optimal class is chosen when at least two of the sub-classifiers have identified the same class. Our method consistently gives a better result with an average 20% improvement in accuracy.

4. CONCLUSION

In this paper, we have shown a discriminative dictionary learning based classification scheme. We have also introduced an information theoretic feature nomination algorithm to choose appropriate features which would be the ma more discriminative feature for the query image. Our method described here chooses the most distinctive query for accurate classification and at the same time does not require comparing the query feature with all the training features. Our experiments show that the algorithm chooses the proper feature for 88.9% cases with at least one of the features having classified the query accurately.

ACKNOWLEDGEMENT

This work is supported in part by DARPA VMR (FA8750-12-C-0181).

REFERENCES

[1] M. Vasconcelos and N. Vasconcelos, "Natural image statistics and low-complexity feature selection," *Pattern Analysis and Machine Intelligence,* vol. 31.2, pp. 228-244, 2009.

[2] Z. Wang, Q. Zhao, D. Chu, F. Zhao and L. J. Guibas, "Select informative features for recognition," in *ICIP*, 2011.

[3] N. Kwak and C. H. Choi, "Input feature selection by mutual information based on Parzen window," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,,* vol. 24(12), pp. 1667-1671, 2002.

[4] H. Peng, F. Long and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on, PAMI,* vol. 27(8), pp. 1226-1238., 2005.

[5] F. Fleuret, "Fast binary feature selection with conditional mutual information," *The Journal of Machine Learning Research ,* vol. 5, pp. 1531-1555., 2004.

[6] B. Epshtein and S. Ullman, "Feature Hierarchies for Object Classification," in *ICCV*, 2005.

[7] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision,* vol. 60.2, pp. 91-110, 2004.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[9] T. Ojala, M. Pietikainen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on ,* vol. 24, no. 7, pp. 971 - 987, 2002.

[10] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry and Y. Ma, "Robust Face Recognition via Sparse Representation," *PAMI, IEEE Transactions on,,* vol. 31(2), pp. 210-227, 2009.

[11] M. Yang, L. Zhang, X. Feng and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *ICCV*, 2011.

[12] Q. Zhang and B. Li, "Discriminative k-svd for dictionary learning in face recognition," 2010, IEEE Conference on Computer Vision and Pattern Recognition.

[13] Z. Jiang, Z. Lin and L. S. Davis, "Label Consistent K-SVD: Learning a Discriminative Dictionary for Recognition,," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 35, no. 11, pp. 2651 - 2664, 2013.

[14] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *Image Processing, IEEE Transactions on,* Vols. 15(12),, pp. 3736-3745., 2006.

[15] E. Parzen, "On estimation of a probability density function and mode.," *Annals of mathematical statistics,* vol. 33(3), pp. 1065-1076., 1962.

[16] L. Fei-Fei, R. Fergus and P. Perona, "Learning generative visual models from few trainig samples an incremental Bayesian approach tested on 101 object categories," in *CVPR, Workshop on Generative-Model based vision*, 2004.

[17] S. Lazebnik, C. Schmid and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006.

[18] L. Breiman, "Bagging predictors," *Machine learning,* vol. 24.2, pp. 123-140, 1996.

[19] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *ICCV*, 2009.

[20] J. Mairal, F. Bach, J. Ponce and G. Sapiro, "Online dictionary learning for sparse coding," *Proceedings of the 26th Annual International Conference on Machine Learning,ACM,* 2009.