Text Classification Using Combined Sparse Representation Classifiers and Support Vector Machines

Neeraj Sharma, Anshu Sharma, Veena Thenkanidiyoor and A. D. Dileep
School of Computing and EE, Indian Institute of Technology Mandi, Mandi, H.P., India
Department of CSE, National Institute of Technology Goa, Ponda, Goa, India
e-mail: addileep@gmail.com

***Abstract*—Text classification is an important task in managing huge repository of textual content prevailing in various domains. In this paper, we propose to use sparse representation classifier (SRC) and support vector machines (SVMs) based classifiers using frequency-based kernels for text classification. We consider term-frequency (TF) representation for a text document. The sparse representation of an example is obtained by using an overcomplete dictionary made up of TF vectors corresponding to all the training documents [1]. We propose to seed the dictionary using principal components of TF vector representation corresponding to training text documents. SVM-based text classifiers use linear kernel or Gaussian kernel on the TF vector representation of documents. TF representation being a non-negative, histogram representation, we propose to build SVM-based text classifiers using frequency-based kernels such as histogram intersection kernel, Chi-square (χ^2) kernel and Hellinger's kernel. It is observed that the examples misclassified by one classifier is correctly classified in another classifier. To take advantage of the various classifiers, we introduce an approach to combine classifiers to improve the performance of text classification. The effectiveness of all the proposed techniques for text classification is demonstrated on 20 Newsgroup Corpus.**

Keywords—sparse representation, sparse representation classifier, support vector machines, frequency-based kernels, Text classification

I. INTRODUCTION

The proliferation of Internet has led to massive amount of information in digital media. The textual content contributes to a large portion of it in addition to multimedia content. For a meaningful usage of such an ocean of information, an efficient mechanism for accessing such a huge repository is essential. Text classification is one important scheme in managing the huge repository. Text classification involves assigning a text document to one of the predefined class or a topic [1]. Usage of a wide variety of classifiers are explored for text classification and every such attempt aims to improve the performance of the classifier used. For example the simple naive Bayes classifier was found to perform well for text classification in [2]. Further, the text classification performance was found to be improved using support vector machines [3]. An important issue in using classifiers such a Bayes classifier, neural networks, nearest neighbor methods is the number of features used for representing a text document. To address this issue, many feature selection methods are considered for text classification [4], [5].

The focus of this work is to design an effective classifier for text classification. We use term-frequency (TF) representation for the text documents. In this paper, we explore a sparse representation classifier (SRC) for text classification. The sparse representation of data is a popular technique in signal processing. The sparse representation classifiers (SRCs) are extensively used in different image and speech processing tasks such as face recognition [6], [7], image classification [8], phonetic classification [9] and speaker verification [10]. To the best of our knowledge, except [1], no other attempt to use sparse representation for text classification is reported. To obtain the sparse representation of data, a dictionary plays an important role. A dictionary $\mathbf{D}$ is a $d \times N_t$ matrix, where each column of the matrix is a d-dimensional training example and N_t is the number of training examples from all the classes. It is required for $\mathbf{D}$ to be an overcomplete dictionary such that the number of examples N_t is much larger than the dimension of each example ($d << N_t$). In signal processing domain, principal components of an example [7], [11] and discrete cosine transform of an example [11] are popularly used for seeding the dictionary. In [1], an overcomplete dictionary $\mathbf{D}$ is constructed using individual training documents with reduced vocabulary. In this work, we propose to use principle components of individual training documents to construct $\mathbf{D}$. We also explore three different rules for classifying the test documents using sparse representation.

The support vector machines (SVMs) are commonly used classifiers for text classification [3]. Conventionally, linear kernel or Gaussian kernel are used in building SVM-based classifiers. However, TF representation corresponding to a document is a non-negative vector (like histogram vector). It is well known in image classification that, when each image is represented as a histogram vector (i.e. bag-of-visual-words), the frequency-based kernels such as histogram intersection kernel (HIK), Chi-square (χ^2) kernel and Hellinger's kernel are more suitable [12]. In this work we propose to explore HIK [13], χ^2-kernel [14] and Hellinger's kernel [12] for text classification using SVMs. To the best of our knowledge these kernels have not been used in the context of text classification.

It is observed that some of the text documents misclassified by SRCs are correctly classified using SVMs-based classifiers and vice versa. To harness the benefit of various classifiers, we explore a simple and novel approach for combining the classifiers. In this approach, we use voting scheme along with posterior probability of a class for combining the classifiers. The effectiveness of all the techniques proposed in this paper are demonstrated for text classification on 20 Newsgroup Corpus [2].

978-1-5090-3488-8/16/$31.00 ©2016 IEEE

The paper makes the following contributions towards exploring SRC and SVMs for text classification. First, we explore seeding the dictionary $\mathbf{D}$ using the principal components of all the training documents. This is in contrast to [1], where all the training documents with reduced vocabulary are used to seed $\mathbf{D}$. Our second contribution is in exploring frequency-based kernels such as HIK, χ^2-kernel and Hellinger's kernel for text classification using SVMs. Third, we introduce a simple technique for combining the classifiers for text classification. We demonstrate its effectiveness in text classification by combining SRCs and SVM-based classifiers.

The rest of the paper is organized as follows. In Section II, sparse representation and classification based on sparse representation are presented. The frequency-based kernels for SVMs are presented in Section III. A simple approach for combining classifiers is presented in Section IV. In Section V, the studies on text classification is presented. The conclusions are presented in Section VI.

II. Sparse Representation Classification

In this section we discuss about the sparse representation and classification using sparse representation.

A. Sparse Representation

Given a feature vector $\mathbf{x} \in \mathbb{R}^d$ and a dictionary $\mathbf{D} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{N_t}]^\top \in \mathbb{R}^{d \times N_t}$ of N_t basis vectors, sparse representation of $\mathbf{x}$ aims to represent it as the linear combination of basis vectors as $\mathbf{x} = \beta_1\mathbf{u}_1 + \beta_2\mathbf{u}_2 + \cdots + \beta_{N_t}\mathbf{u}_{N_t}$. Here $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots \beta_{N_t}]^\top$ is the coefficient vector where β_n is the coefficient associated with the basis vector $\mathbf{u}_n$. The sparseness in representation is achieved by ensuring only a small fraction of elements in $\boldsymbol{\beta}$ to be non-zero. The problem of obtaining sparse representation, $\boldsymbol{\beta}$ of $\mathbf{x}$ can be formulated as

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_0 \text{ such that } \mathbf{x} = \mathbf{D}\boldsymbol{\beta} \tag{1}$$

Here $\|\boldsymbol{\beta}\|_0$ is the l_0-norm, which counts the number of nonzero entries in $\boldsymbol{\beta}$. However, the minimization of l_0-norm is an NP hard problem [15]. Recent developments in sparse representation [16] indicate that if the solution $\boldsymbol{\beta}$ is sparse enough, then l_0-norm in (1) can be replaced with an l_1-norm as

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_1 \text{ such that } \mathbf{x} = \mathbf{D}\boldsymbol{\beta} \tag{2}$$

The necessary condition for working with sparse representation based methods is that the dictionary $\mathbf{D}$ should be overcomplete, i.e., $d << N$

B. Classfication Based on Sparse Representation

In the classification problem, training examples of all the classes act as basis vectors in $\mathbf{D}$. Now the dictionary matrix can be seen as $\mathbf{D} = [D_1, D_2, \ldots, D_M]^\top$, where M is the number of classes. Here, $D_m = [\mathbf{x}_{m1}, \mathbf{x}_{m2}, \ldots, \mathbf{x}_{mN_m}]^\top$ where $\mathbf{x}_{mn}$, for $n = 1, 2, \ldots, N_m$, are the training examples of the mth class. A test example belonging to mth class can be seen as the linear combination of training examples of mth class. For this example, the coefficients (β) associated with the training examples of mth class is non-zero and all the remaining coefficients are zero. The sparse vector $\boldsymbol{\beta}$, for the example is denoted as $\boldsymbol{\beta} = [0, \ldots, 0, \beta_{m1}, \beta_{m2}, \ldots, \beta_{mN_m}, 0, \ldots, 0]^\top$. Ideally the optimal $\boldsymbol{\beta}$ for a test example should be sparse and non-zero for the coefficients associated with the training examples belonging to the class of test example.

The main issue in considering sparse representation based methods for text classification is in having overcomplete dictionary matrix $\mathbf{D}$. The dimension of TF vector representation considered for each text document depends on the vocabulary size. Typically the size of vocabulary is larger than the number of training examples in a data set. When the TF vector representation of training examples are used as basis vector, the resulting $\mathbf{D}$ matrix becomes under complete, i.e., $d >> N_t$. This violates the necessary condition that $\mathbf{D}$ must be overcomplete. To address this issue, a reduced vocabulary ($d << N_t$) is considered in [1]. In this work, we are not reducing the vocabulary size. Instead, we propose to explore the principle component representation of TF vectors corresponding to documents to seed the dictionary. This approach is popular in sparse representation based face recognition [6]. In this work, we consider the leading principal components to ensure that the number of principal components to be less than the number of training documents.

We explore following 3 different classification rules [1] to assign a class label to a test example.

1) Maximum support: Ideally, all non-zero entries of $\boldsymbol{\beta}$ should correspond to the training examples in $\mathbf{D}$ of the same class as the test example. In this ideal situation, a test example will be assigned with the class label of the training example which has the largest values in $\boldsymbol{\beta}$.

2) Maximum l_2 support: In practice, elements of $\boldsymbol{\beta}$ corresponding to other than the class of test example could also be non-zero. To consider this fact, we compute the l_2-norm for all the $\boldsymbol{\beta}$ entries for a class and choose the class with largest l_2-norm as class label to the test example [10]. Let $\boldsymbol{\delta}_m(\boldsymbol{\beta})$ be a vector whose entries are the β values for the class m. Then assignment of the class label to a test example is given as

$$\text{class label} = \max_m(\|\boldsymbol{\delta}_m(\boldsymbol{\beta})\|_2); \text{ for all } m = 1, 2, \ldots, M \tag{3}$$

3) Minimum residual error: Let $\mathbf{x}$ be the test example and $\mathbf{D}\boldsymbol{\beta}$ gives the reconstruction of $\mathbf{x}$. The difference between $\mathbf{x}$ and its reconstruction gives the residual error. Since the β belonging to other classes can also be non-zero, $\|\mathbf{x} - \mathbf{D}\boldsymbol{\delta}_m(\boldsymbol{\beta})\|_2$ gives the residual error for the class m [6]. The class with smallest residual error will be considered as class label for the test example.

$$\text{class label} = \min_m(\|\mathbf{x} - \mathbf{D}\boldsymbol{\delta}_m(\boldsymbol{\beta})\|_2); \text{ for all } m = 1, 2, \ldots, M \tag{4}$$

In the next section we present the frequency-based kernels for support vector machines.

III. Frequency-based Kernels for SVM

In this section we present the frequency-based kernels for the SVMs that are more suitable when the examples are represented as non-negative vectors i.e., histogram vectors. The frequency-based kernels are successfully used in image classification when each image is represented in bag-of-visual-words (BOVW) representation [12]. The BOVW representation

of an image is basically a histogram vector representation. Histogram intersection kernel (HIK) [13], Chi-square (χ^2) kernel [14] and Hellinger's kernel [12] are some of the popular frequency-based kernels used in image classification. The TF representation of a document is also a non-negative vector (like histogram vector) representation. Hence, we propose to use HIK, χ^2-kernel and Hellinger's kernel for text classification using SVMs

A. Histogram Intersection Kernel (HIK)

Let $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^\top$ and $\mathbf{x}_j = [x_{j1}, x_{j2}, \ldots, x_{jd}]^\top$ be the TF vector representation corresponding to the two documents. Here, d is the size of vocabulary. The number of matches in the qth bin is given by histogram intersection function [17], defined as follows:

$$s_q = \min\,(x_{iq}, x_{jq}) \tag{5}$$

An HIK is computed as the total number of matches and is given by,

$$K_{\text{HIK}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{q=1}^{d} s_q \tag{6}$$

B. Chi-square (χ^2) Kernel

The Chi-square (χ^2) kernel [14] is also computed in the similar lines as HIK. Here, the number of matches in the qth bin is given by

$$s_q = \frac{2(x_{iq}x_{jq})^2}{x_{iq} + x_{jq}} \tag{7}$$

A χ^2 kernel is computed as the total number of matches and is given by,

$$K_{\chi^2}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{q=1}^{d} s_q \tag{8}$$

C. Hellinger's kernel

The Hellinger's kernel [12] is also computed in the similar lines as HIK. Here, the number of matches in the qth bin is given by

$$s_q = \sqrt{x_{iq}x_{jq}} \tag{9}$$

The Hellinger's kernel is computed as the total number of matches and is given by,

$$K_{HK}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{q=1}^{d} s_q \tag{10}$$

In the next section, we present an approach to combine the SRCs and SVM-based classifiers.

IV. Combining Classifiers

It is observed that some of the text documents misclassified by SRCs are correctly classified using SVMs and vice versa. To get the benefit of various classifiers, we propose a voting based approach for combining the classifiers. Let M denote the number of classes and K corresponds to the number of classifiers. Every classifier λ_k for $k = 1, 2, \ldots, K$ generates a vote v_k for a test example. Here, $v_k \in \{1, 2, \ldots, M\}$. A class label to a test example is assigned as

$$\text{class label} = \text{mode}(v_1, v_2, \ldots, v_k, \ldots, v_K) \tag{11}$$

where mode(.) corresponds to the value that occurs maximum number of times. This approach may fail if there is no agreement between classifiers. This may happen when no two classifiers vote for the same class for a test example. For such a test example, class label is assigned based on the posterior probability of a class. Every classifier computes a posterior probability of a class for the test example. Class label for a test example is assigned based on the class with the maximum posterior probability in any of the classifier.

In the next section we present our studies on text classification to evaluate the techniques introduced this paper.

V. Experimental Studies on Text Classification

In this section we first present the details of the data set used in the studies and features considered to represent documents. Next we present and discuss the results of the studies on text classification.

The 20 Newsgroup corpus [2] is used for evaluating the techniques introduced in this paper for text categorization. This corpus consists of 18,774 text documents divided into 20 different newsgroup classes. Among them 60% of the documents (i.e. 11,269) are used for training the models and remaining 40% (i.e. 7,505) of the documents are used for test. The text document categorization accuracy presented is the classification accuracy obtained for the test examples. In this study, we considered term-frequency (TF) as a feature. We have considered 53,975 terms (or words) from all the training documents as the vocabulary. The frequency of occurrence of each of the vocabulary term in a text document is computed. Every document is represented as a 53,975-dimensional TF vector.

First, we present the studies on text classification using sparse representation classifier (SRC). SRC requires the construction of dictionary matrix $\mathbf{D}$ by considering at every column, the training examples represented as TF vector. For the 20 Newsgroup corpus, the number of documents ($Nt = 11,269$) is much smaller than the dimension of the TF vector representation for documents ($d = 53,975$). For the successful application of SRC, we need $d << N_t$. To comply with this requirement, we applied principal component analysis (PCA) on the document vectors. This involves projecting a document vector along the eigen vectors corresponding to the leading eigen values. Then we used principal component representation in $\mathbf{D}$. The performance of SRC is analyzed by building $\mathbf{D}$ with the number of principal components $\hat{d}$ taking the values from 1,000 to 11,000 with the increment of 1,000. The classification accuracy of SRC for the varied number of principal components is presented in Figure 1. The Figure 1 also compares the performance of SRC with that of Gaussian kernel (GK) based SVM classifier for text classification, where each document is represented with principal components. It is seen that the SRC using $\mathbf{D}$ matrix with $\hat{d}$=6,000 give the best performance for text classification. It is also observed that SRC using maximum l_2 support as classification rule performs significantly better than that of the SRCs using maximum

support and minimum residual error as classification rules. The best classification accuracy of SRC using maximum l_2 support is observed to be **78**.**78**% for $\hat{d}$=6,000, which is comparable with that of the best accuracy (78.80%) reported in [1]. In [1], the vocabulary size is reduced to 11,000 and TF vector reposentation is obtained using this vocabulary. It is also observed that text classification using SRC is marginally better than that of the Gaussian kernel based SVM classifier.

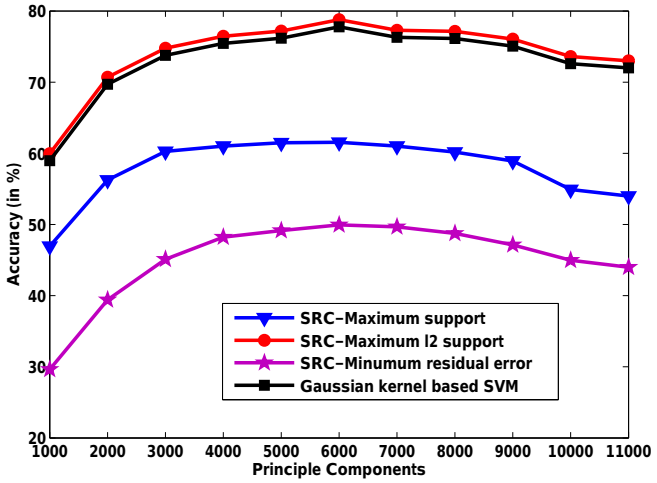

Figure 1. Classification accuracy (in %) using SRC with maximum support rule, maximum l_2 support rule & minimum residual error rule and Gaussian kernel based SVM for varied number of principal components.

Next, we present the studies on text classification using SVM classifier using frequency-based kernel. In this study we considered TF vector representation with dimension $d = 53,975$ for each text document. We consider SVMTorch [18] tool to build the SVM-based classifiers. In this study, one-against-the-rest approach is considered for 20-class text document classification. The value of trade-off parameter, C in SVM is chosen empirically. The classification accuracies for the SVM-based classifier using frequency-based kernels such as histogram intersection kernel (HIK), Chi-square (χ^2) kernel and Hellinger's kernel are given in Table I for text classification. It also compares the classification accuracies with that of the Gaussian kernel based SVM and SRC. It is seen that SVM using Hellinger's kernel performed marginally better than that of SVMs using HIK and χ^2-kernel. It is also seen that the performance of the SVM-based classifiers using frequency-based kernels is comparable with the that of the SRC and marginally better than the Gaussian kernel based SVMs. This shows that the SVM-based classifiers using frequency-based kernels perform equally better.

We noticed that some of the text documents misclassified by SRCs are correctly classified using SVMs and vice versa. We also observed the similar behavior with in the SRCs with different decision rules as well as with in the SVMs with different kernels. This motivated us to combine the classifiers. We considered a simple rule for combining the decisions of classifiers and is based on the ground truth corresponding to every test example. Let K be the number of classifiers and observe the decisions made by each of them. If any one among K classifiers assigned a correct label to a text example, then that will be the class label for that text example. Table II present the classification accuracies after combining the different SRCs and SVM-based classifiers. This table presents the classification accuracy obtained after (i) combining SRCs with 3 different decision rules (mentioned in Section II-B), (ii) combining SVM-based classifiers with Gaussian kernels, HIK, χ^2-kernel and Hellinger's kernel, and (iii) combining all the three SRCs and four SVM-based classifiers. It is seen that the text classification accuracy is improved significantly. It is observed that when the SRCs with 3 different decision rules are combined, about 5% of new examples get classified correctly. Similarly about 7% of new examples get classified correctly when SVM-based classifiers using 4 different kernels are combined and about 8% of new examples get classified when all the SRCs and SVM-based classifiers are combined.

TABLE I. CLASSIFICATION ACCURACY (CA) (IN %) OF THE SRC AND SVM-BASED CLASSIFIERS USING HIK, χ^2-KERNEL, HELLINGER'S KERNEL AND GAUSSIAN KERNEL FOR TEXT CLASSIFICATION. HERE TF IS TERM FREQUENCY AND PCA IS PRINCIPAL COMPONENT ANALYSIS

Representation	Classification model		CA
TF	SVM using	HIK	77.56
		χ^2-kernel	76.89
		Hellinger's kernel	77.96
		GK	77.30
Principal component based	SVM using	GK	77.70
	SRC	-	78.78

TABLE II. CLASSIFICATION ACCURACY (CA) (IN %) OF THE COMBINED SRCS AND SVM-BASED CLASSIFIERS FOR TEXT CLASSIFICATION

Combining SRCs using 3 different decision rules	Combining SVMs using 4 different kernels	Combining all the SRCs and SVMs
82.65	83.93	85.48

The main drawback of the above mentioned technique is that, it requires the presence of ground truth for combining the classifiers. However, the above study give the evidence that there is scope for combining the classifiers in text classification. Next we evaluate the approach introduced in the Section IV for combining the decisions of the classifiers. The approach is used for combining the decisions from seven classifiers (SRCs with 3 different decision rules + SVM-based classifiers with 4 different kernels). Table III compares the classification accuracy after combining the decisions from seven classifiers with the accuracy obtained using SRC and SVM-based classifiers. It is seen that the text classification accuracy is improved as compared to that of any single classifiers. It is observed that around 3% of new examples get classified correctly when seven classifiers are combined using the proposed approach. Though this improvement is not as high as in Table II, there is scope for improvement in the method for combining the classifiers.

VI. CONCLUSIONS

Approaches to text casification using sparse representation classifiers (SRC) and support vector machines (SVMs) are explored in this paper. Sparse representation of a text document is obtained as a linear combination of documents in the

training set. Sparseness is assured by making most of the coefficients to be zero in the linear combination. To obtain sparse representation, an overcomplete dictionary (number of columns being much larger than the number of rows) made up of training documents is used. In text classification, typically dictionaries turn out to be under complete. To overcome this issue, we built an overcomplete dictionary using the principal components based representation corresponding to the training documents. SVM-based classifiers for text classification usually use linear or GK on TF vector representation of documents. The TF vector being non-negative histogram vector, we considered frequency based kernels, histogram intersection kernel, χ^2-kernel and Hellinger's kernel for building SVMs. It was observed that examples misclassified in one classifier get correctly classified in the other classifier. To harness the capabilities of the different classifiers, we explored a method for combining the classifiers based on voting scheme. The effectiveness of the proposed techniques is demonstrated using 20 Newsgroup Corpus.

The misclassification observed in 20 Newsgroup corpus can be attributed to the presence of highly confusing classes. It is also observed that the confusing classes are also semantically related. These semantically related classes can be grouped into a single class and a hierarchical classifier built using SVMs and SRC is expected to perform better. Possibly the classifier built using the semantic information may also improve the performance of text classification. The approach explored for combining the classifiers can be extended to any type of the classifiers. Better approaches for combining the classifiers need to be explored.

TABLE III. CLASSIFICATION ACCURACY (CA) (IN %) OF THE COMBINED SRCS AND SVM-BASED CLASSIFIERS FOR TEXT CLASSIFICATION USING THE PROPOSED APPROACH FOR COMBINING CLASSIFIERS AND THE BEST PERFORMANCE OBTAINED USING SRCS AND SVM-BASED CLASSIFIERS.

SRC	SVM	Combined classifier
78.78	77.96	81.83

REFERENCES

[1] Tara N Sainath, Sameer Maskey, Dimitri Kanevsky, Bhuvana Ramabhadran, David Nahamoo, and Julia Hirschberg, "Sparse representations for text categorization.," in *Proceedings of INTERSPEECH*, 2010, vol. 10, pp. 2266–2269.

[2] Andrew McCallum and Kamal Nigam, "A comparison of event models for naive Bayes text classification," in *Proceedings of AAAI-98 workshop on learning for text categorization*, 1998, vol. 752, pp. 41–48.

[3] Thorsten Joachims, "Text categorization with suport vector machines: Learning with many relevant features," in *Proceedings of the 10th European Conference on Machine Learning (ECML'98)*. ACM, 1998, pp. 137–148.

[4] Anirban Dasgupta, Petros Drineas, Boulos Harb, Vanja Josifovski, and Michael W. Mahoney, "Feature selection methods for text classification," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07)*, 2007, pp. 230–239.

[5] George Forman, "An extensive empirical study of feature selection metrics for text classification," *Journal of Machine Learning Research*, vol. 3, pp. 1289–1305, 2003.

[6] John Wright, Allen Y. Yang, Arvind Ganesh, S. Shankar Sastry, and Yi Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, February 2009.

[7] T. Shejin and Anil Kumar Sao, "Significance of dictionary for sparse coding based face recognition," in *Proceedings of 11th International Conference of the Biometrics Special Interest Group*, Darmstadt, Germany, September 2012, pp. 1–6.

[8] Xiao-Tong Yuan and Shuicheng. Yan, "Visual classification with multitask joint sparse representation," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4349–4360, October 2012.

[9] Tara N Sainath, Avishy Carmi, Dimitri Kanevsky, and Bhuvana Ramabhadran, "Bayesian compressive sensing for phonetic classification," in *Proceedings of IEEE International Conference onAcoustics Speech and Signal Processing (ICASSP 2010)*, 2010, pp. 4370–4373.

[10] Jia Min Karen Kua, Julien Epps, and Eliathamby Ambikairajah, "i-Vector with sparse representation classification for speaker verification," *Speech Communication*, vol. 55, no. 5, pp. 707 – 720, 2013.

[11] M. Elad, *Sparse and Redundant Representations - From Theory to Applications in Signal and Image Processing*, Springer, 2010.

[12] Ken Chatfield, Victor S Lempitsky, Andrea Vedaldi, and Andrew Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *Proceedings of the 22nd British Machine Vision Conference (BMVC 2011)*, Kingston, UK, September 2011, pp. 76.1–76.12.

[13] Jan C. van Gemert, Cor J. Veenman, Arnold W.M. Smeulders, and Jan-Mark Geusebroek, "Visual word ambiguity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 17, pp. 1271–1283, July 2010.

[14] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010, pp. 3539–3546.

[15] Edoardo Amaldi and Viggo Kann, "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems," *Theoretical Computer Science*, vol. 209, no. 12, pp. 237 – 260, 1998.

[16] David L Donoho, "For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[17] Michael J. Swain and Dana H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, November 1991.

[18] R. Collobert and S. Bengio, "SVMTorch: Support vector machines for large-scale regression problems," *Journal of Machine Learning Research*, pp. 143–160, 2001.