Article:
Chen, R., Hawes, M. and Mihaylova, L. orcid.org/0000-0001-5856-2223 (2020) A nonparametric Bayesian compressive sensing classification. Journal of Advances in Information Fusion, 15 (1). pp. 57-70. ISSN 1557-6418

A Non-parametric Bayesian Compressive Sensing Classification

Ruilong Chen, Matthew Hawes and Lyudmila Mihaylova

Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield S1 3JD, United Kingdom,
Email: chen8131928@gmail.com, hawes.matthewblair@gmail.com, l.s.mihaylova@sheffield.ac.uk

***Abstract*—This paper presents a novel non-parametric back-propagation Bayesian compressive sensing (BBCS) classification approach. While the state-of-the-art parametric classifiers such as logistic regression require model training and can result in inadequate models, the developed approach does not require model training. It is combined with a column-based subspace sampling process and it can deal efficiently with uncertainties and highly computational tasks. Validation on a publicly available vehicle logo dataset shows that the proposed classifier can achieve up to 98% recognition accuracy as compared with the state-of-the-art non-parametric classifiers. Compared with the generic Bayesian compressive sensing classification, the proposed approach decreases the mean number of misclassifications by 87% and with 68% reduction of the computational time. The robustness of the BBCS approach is demonstrated over scene recognition tasks, and its outperformance over the AlexNet convolutional neural networks algorithm is demonstrated in noisy conditions. The proposed BBCS approach is generic and can be used in different areas, for example, it has shown robustness over the CIFAR-10 dataset.**

I. Introduction

A number of parametric classifiers such as the linear Support Vector Machine (SVM) [1–4] and Logistic Regression (LR) [5] have been developed for Vehicle Logo Recognition (VLR) and Traffic Scene Recognition (TSR). Deep learning models such as Convolutional Neural Networks (CNN) and capsule networks have been applied to VLR [6, 7]. These parametric classifiers assume a functional distribution of the data [8]. The relationship between the label and the input data is modeled using a fixed number of parameters. An advantage of parametric classifiers is that once the number of parameters are determined, it would not change later as non-parametric methods do. However, in practice, parametric classifiers could result in an inadequately trained model due to inappropriate assumptions of prior distributions, leading to inappropriate predictions in the testing phase [8, 9].

On the contrary, non-parametric classifiers do not make assumptions about the distribution representing the data [8]. They do not have a model with a fixed number of parameters. Instead, the number of parameters increases with the size of the training dataset [10]. This in turn increases the computational complexity.

The K-Nearest Neighbor (KNN) approach is a commonly used non-parametric approach which is often used for classification [11, 12]. However, the KNN approach is not robust to outliers and to data with high dimensionality. This is because the shortest distance is not necessarily the best match to the testing data, especially when the number of training data are limited [8, 13]. Besides, the KNN approach has been shown to be vulnerable to noise effects [5].

A non-parametric classification approach based on sparse representation proposed by Wright et al. [14] has proven to be more accurate than the linear SVM and the KNN classifier for face recognition. The Sparse Representation Classifier (SRC) [14] assumes that the testing data can be represented as a linear combination of the training dataset. A weight vector is generated with each element representing a corresponding coefficient in the linear combination. By splitting the weights according to their associated classes (with the remaining set to be zero valued), the weights in the correct class should reconstruct the original data with a minimum error. However, the high computational costs of the SRC can be a problem. In addition, the SRC works only when the system is underdetermined [15]. In practice, this criterion cannot be met when there is a lack of training data.

Recently the Bayesian Compressive Sensing (BCS) [16] approach has been efficiently applied to synthetic aperture radar target classification [17], image reconstruction [18, 19] and phonetic classification [20]. The Bayesian approach could potentially provides an alternative to the l_1-norm minimization for optimizing the linear combination coefficients required for the classification framework. Similarly to Zhou et al. [21], by comparing the magnitudes of the coefficients, the testing data can then be classified by assigning it to the class whose coefficients have the highest l_2-norm magnitude.

The methods proposed in [22, 23] map the data into a reduced dimensional space, as the Principle Component Analysis (PCA). However, these new latent spaces are different

from the original space and make the original data difficult to interpret. To combat this issue a column-based subspace sampling data representation can be used [24–26]. In this case it is still possible to work in the original space, just with fewer data points.

In order to cope with various sources of uncertainties that many of the existing classification algorithms face, this paper proposes a new solution that provides robustness to insufficient training data and to noises. The key contributions of this work can be summarized as follows:

I) A new Backpropagation Bayesian Compressive Sensing (BBCS) classifier is developed which represents efficiently the data and solves the classification problem as an optimization problem. The Euclidean distance between the constructed testing data and the original testing data is minimized. This process increases the recognition accuracy.

The BBCS incorporates a data reduction process that further decreases the computational costs. The column-based subspace sampling representation selects informative data points from the dataset. Compared with the PCA which transforms the original data into a new latent space, the column-based subspace sampling method chooses the best data directly from the original space. This process significantly decreases the computational costs and facilitates the interpretation in this reduced dimensional space.

II) The developed BBCS approach is validated and evaluated over noisy data and compared with state-of-the-art non-parametric classifiers: the KNN algorithm, the SRC and the BCS algorithm. The BBCS is more robust than KNN classifier. Compared with the BCS, the proposed approach decreases the mean number of misclassifications by 87% and by reducing the computational cost compared with the SRC algorithm.

The rest of this paper is organized as follows. Section II introduces the general sparse representation classification framework. Section III.A presents the BCS approach. Section III.B introduces the developed backpropagation BCS classifier approach and the derivation of the theoretical relationships. Section III.C introduces the column-based subspace sampling method. Section IV presents performance validation of the BBCS and discussions of the results. Section V summarizes the results. The *Appendix* contains the full derivation of the marginal likelihood function and its maximization.

II. Classification framework based on sparse representation

The SRC, BCS classifier and BBCS classifier assume that the testing data $\mathbf{x}^* \in \mathbb{R}^{M\times 1}$ can be represented as a linear combination of the training samples $\mathbf{X} \in \mathbb{R}^{M\times N}$ where M is the length of the vector data and N gives the number of entries in the training dataset. When applying to images, each image is represented by an image feature vector rather than by pixels of the raw image. Therefore, M refers to the length of the feature vector representing the image. Feature-based methods such as the Scale Invariant Feature Transform (SIFT) [27] and CNN [28] can represent an image using a vector rather than a matrix representation.

A testing image denoted by image feature $\mathbf{x}^*$ is represented with the linear model:

$$\mathbf{x}^* = \mathbf{X}\mathbf{w} + \mathbf{z}, \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^{N\times 1}$ is a weight vector controlling the contribution of each image feature in the training dataset to the linear combination representing the testing image feature, $\mathbf{z} \in \mathbb{R}^{M\times 1}$ is a bounded noise term with $||\mathbf{z}||_2 \leqslant \epsilon$, $||\cdot||_2$ is the l_2-norm and ϵ is a small positive constant. The solution to (1), $\mathbf{w}$, is obtained by minimizing the l_2-norm:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}}(||\mathbf{w}||_2), \quad s.t \;\; ||\mathbf{x}^* - \mathbf{X}\mathbf{w}||_2 \leqslant \epsilon, \tag{2}$$

where $\hat{\mathbf{w}} \in \mathbb{R}^{N\times 1}$ is the estimated weight vector. However, when $N > M$, equation (1) corresponds an under-determined system and there is no unique solution by using conventional methods [14, 29].

The SRC classification method [14] assumes that a testing image feature can be sufficiently represented by a dictionary for its corresponding class. Therefore, the solution is naturally sparse as coefficients for unrelated classes are zero valued. For instance, if there are 20 classes, only approximately 5% of the coefficients in $\hat{\mathbf{w}}$ will have non-zero values [14]. In fact, the sparser the recovered $\mathbf{w}$ is, the easier it is to accurately classify the testing image feature $\mathbf{x}^*$ [14]. This motivates the use of the l_0-norm to find the sparest solution for $\mathbf{w}$ in equation (1).

However, l_0-norm minimization is an NP hard problem. Instead, an l_1-norm minimization is typically used as an approximation [15, 30, 31], giving:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}}(||\mathbf{w}||_1), \quad s.t \;\; ||\mathbf{x}^* - \mathbf{X}\mathbf{w}||_2 \leqslant \epsilon. \tag{3}$$

The solution to the l_1-minimization in equation (3) can be found by linear programming methods such as the basis pursuit [32] or the orthogonal matching pursuit [33] methods. The solution to equation (1) gives the optimal $\mathbf{w}$ for classification purposes in the SRC [14].

III. Bayesian Compressive Sensing

The BCS method [16] provides an alternative to the l_1-norm minimization method by incorporating prior knowledge within the Bayesian framework. Since the testing image feature can be represented as a linear combination (1) of the training images, the relative importance of each training image feature is controlled by the weight vector $\mathbf{w}$. The vector $\mathbf{w}$ can be

separated into $\mathbf{w}_v$ and $\mathbf{w}_e$, where $\mathbf{w}_v$ contains the significant weights and $\mathbf{w}_e$ the remaining negligible weights. Hence, $\mathbf{w} = \mathbf{w}_v + \mathbf{w}_e$ and equation (1) can be written as:

$$\mathbf{x}^* = \mathbf{X}\mathbf{w}_v + \mathbf{X}\mathbf{w}_e + \mathbf{z}. \tag{4}$$

Both $\mathbf{X}\mathbf{w}_e$ and $\mathbf{z}$ can be approximated as zero-mean Gaussian noises [16], allowing equation (4) to be written as:

$$\mathbf{x}^* = \mathbf{X}\mathbf{w}_v + \mathbf{n}, \tag{5}$$

where $\mathbf{n} = \mathbf{X}\mathbf{w}_e + \mathbf{z}$. The variance of $\mathbf{n}$ is then given by $\mathbf{\Sigma}_n = \sigma^2\mathbf{I}_M$, where $\mathbf{I}_M$ is an identity matrix of size M $\times$ M. Note that each entry in $\mathbf{n}$ has the same variance σ^2 and hence the likelihood function can be given by:

$$p(\mathbf{x}^*|\mathbf{w}, \sigma^2) = (2\pi\sigma^2)^{-\frac{M}{2}} exp\left\{-\frac{||\mathbf{x}^* - \mathbf{X}\mathbf{w}||_2^2}{2\sigma^2}\right\}, \tag{6}$$

rather than in the standard multivariate form which includes the covariance matrix $\mathbf{\Sigma}_n$. In (6) and in the following equations the subscript v of $\mathbf{w}$ is dropped for conciseness.

The elements of $\mathbf{w}$ are assumed to have a zero mean Gaussian distribution. This is given by:

$$\begin{aligned} p(\mathbf{w}|\boldsymbol{\alpha}) &= \prod_{i=1}^{N} N(w_i|0, \alpha_i^{-1}) \\ &= \prod_{i=1}^{N} (2\pi\alpha_i^{-1})^{-\frac{1}{2}} exp\left\{-\frac{1}{2}\alpha_i w_i^2\right\} \\ &= (2\pi)^{-\frac{N}{2}} |\mathbf{A}|^{\frac{1}{2}} exp\left\{-\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{w}\right\}, \end{aligned} \tag{7}$$

where $\mathbf{A} = diag(\alpha_1, \alpha_2, \cdots, \alpha_N)$ and $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_N]^{\mathrm{T}}$, α_i is a precision value and $|\cdot|$ denotes the determinant. Furthermore, Gamma hierarchical priors are considered over α_i and σ^2:

$$p(\boldsymbol{\alpha}) = \prod_{i=1}^{N} Gamma(\alpha_i|a, b), \tag{8}$$

$$p(\sigma^2) = Gamma(\sigma^2|c, d), \tag{9}$$

where a, b, c and d are shape and scale parameters.

The overall prior over $\mathbf{w}$ can be evaluated by marginalizing over the hyperparameters $\boldsymbol{\alpha}$:

$$p(\mathbf{w}|a, b) = \prod_{i=1}^{N} \int_0^{\infty} N(w_i|0, \alpha_i^{-1}) Gamma(\alpha_i|a, b) d\alpha_i. \tag{10}$$

Since the prior of $\mathbf{w}$ is assumed to be a zero-mean Gaussian distribution which conjugates to a Gamma prior, the probability density $p(\mathbf{w}|a, b)$ corresponds to a Student's t-distribution [34]. This achieves sparsity as the Student's t-distribution can be strongly peaked at $w_i = 0$ with appropriate choices of a and b [16, 34].

Combining the likelihood function and the prior given by equations (6) and (7), respectively, the posterior distribution of the weights can be found from:

$$p(\mathbf{w}|\mathbf{x}^*, \boldsymbol{\alpha}, \sigma^2) = \frac{p(\mathbf{x}^*|\mathbf{w}, \sigma^2)p(\mathbf{w}|\boldsymbol{\alpha})}{p(\mathbf{x}^*|\boldsymbol{\alpha}, \sigma^2)}. \tag{11}$$

As the likelihood function and prior are both Gaussian, the posterior distribution over $\mathbf{w}$ is also a Gaussian distribution:

$$\begin{aligned} &p(\mathbf{w}|\mathbf{x}^*, \boldsymbol{\alpha}, \sigma^2) = N(\mathbf{w}|\boldsymbol{\mu}, \mathbf{\Sigma}), \\ &= (2\pi)^{-N/2} |\mathbf{\Sigma}|^{-1/2} exp\left\{-\frac{1}{2}(\mathbf{w} - \boldsymbol{\mu})^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{w} - \boldsymbol{\mu})\right\}, \end{aligned} \tag{12}$$

where the mean vector and covariance matrix are given respectively by:

$$\boldsymbol{\mu} = \sigma^{-2}\mathbf{\Sigma}\mathbf{X}^{\mathrm{T}}\mathbf{x}^* \tag{13}$$

and

$$\mathbf{\Sigma} = (\mathbf{A} + \sigma^{-2}\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}. \tag{14}$$

Notice that $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$ are dependent on σ^2 and $\boldsymbol{\alpha}$. Therefore, the goal is to find the posterior probability density function over all the unknown parameters given the training image features and the testing image feature. This means finding the values for $\mathbf{w}$, $\boldsymbol{\alpha}$ and σ^2 which maximize the following posterior probability density function:

$$p(\mathbf{w}, \boldsymbol{\alpha}, \sigma^2|\mathbf{x}^*) = p(\mathbf{w}|\mathbf{x}^*, \boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\alpha}, \sigma^2|\mathbf{x}^*). \tag{15}$$

Finding the optimal $\mathbf{w}$, $\boldsymbol{\alpha}$ and σ^2 involves two steps. Firstly, for the current values of $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$, the values of $\boldsymbol{\alpha}$ and σ^2 are calculated to maximize $p(\boldsymbol{\alpha}, \sigma^2|\mathbf{x}^*)$. Then these values are substituted to re-evaluate $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$. This process is then repeated until a convergence criterion is met. In the first step $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$ are fixed then maximizing equation (15) is equivalent to maximizing:

$$p(\boldsymbol{\alpha}, \sigma^2|\mathbf{x}^*) = \frac{p(\mathbf{x}^*|\boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\alpha})p(\sigma^2)}{p(\mathbf{x}^*)}, \tag{16}$$

where the denominator is independent of $\boldsymbol{\alpha}$ and σ^2. Therefore, only $p(\mathbf{x}^*|\boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\alpha})p(\sigma^2)$ has to be maximized. Furthermore, by selecting a, b, c and d to be small positive values there are flat, uninformative priors over α and σ^2 [34]. Maximizing equation (16) is approximately equal to maximizing the marginal likelihood:

$$p(\mathbf{x}^*|\boldsymbol{\alpha}, \sigma^2) = \int p(\mathbf{x}^*|\mathbf{w}, \sigma^2)p(\mathbf{w}|\boldsymbol{\alpha})d\mathbf{w}, \tag{17}$$

with $p(\mathbf{x}^*|\mathbf{w}, \sigma^2)$ and $p(\mathbf{w}|\boldsymbol{\alpha})$ being given in equations (6) and (7), respectively. The full derivation of the marginal likelihood function is given in *Appendix A*.

Equation (17) is a convolution of two zero-mean Gaussians and the logarithm of the result gives:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\alpha}, \sigma^2) &= ln\left(p(\mathbf{x}^*|\boldsymbol{\alpha}, \sigma^2)\right) \\ &= ln\left(N(\mathbf{x}^*|0, \mathbf{C})\right) \\ &= -\frac{1}{2}\left(Mln(2\pi) + ln|\mathbf{C}| + \mathbf{x}^{*T}\mathbf{C}^{-1}\mathbf{x}^*\right), \end{aligned} \quad (18)$$

where the $M \times M$ matrix $\mathbf{C}$ is given by:

$$\mathbf{C} = \sigma^2\mathbf{I}_M + \mathbf{X}\mathbf{A}^{-1}\mathbf{X}^{\mathrm{T}}. \quad (19)$$

A type-II maximum likelihood approximation is used to estimate $\boldsymbol{\alpha}$ and σ^2 [34], which gives:

$$\alpha_i^{new} = \frac{1 - \alpha_i\Sigma_{ii}}{\mu_i^2}, \quad (20)$$

$$(\sigma^{new})^2 = \frac{||\mathbf{x}^* - \mathbf{X}\boldsymbol{\mu}||_2^2}{M - \sum_i^N(1 - \alpha_i\Sigma_{ii})}, \quad (21)$$

where Σ_{ii} is the i-th diagonal element of $\boldsymbol{\Sigma}$ in equation (14). The parameters $\boldsymbol{\alpha}$ and σ^2 are functions of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, while $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are functions of $\boldsymbol{\alpha}$ and σ^2. This leads to an iterative algorithm to update each variable until a convergence criterion has been met. The derivation of the update equations in (20) and (21) is provided in *Appendix B*.

IV. The Proposed Backpropagation Bayesian Compressive Sensing Classifier and Column-based Subspace Sampling

A. Backpropagation Bayesian Compressive Sensing Classifier

Given that the training images in $\mathbf{X}$ belong to K classes, where the class label $i \in \{1, 2, \cdots, K\}$, the training image features can be separated according to their labels. This gives $\mathbf{X} = [\mathbf{X}^1, \mathbf{X}^2, \cdots, \mathbf{X}^i, \cdots, \mathbf{X}^K]$, where $\mathbf{X}^i$ contains all of the training image features belonging to the i^{th} class. Suppose that there are n_i samples in the i^{th} class, then all of the training image features in the i^{th} class are given by $\mathbf{X}^i = [\mathbf{x}_1^i, \mathbf{x}_2^i, \cdots, \mathbf{x}_{n_i}^i]$. Notice, this process only separates the training image features by their labels, the total number of training image features does not change. Hence, $\sum_i^K n_i = N$.

Therefore the original testing image feature can be reconstructed by using the estimated weight vector $\hat{\mathbf{w}}$:

$$\tilde{\mathbf{x}}^* = [\mathbf{X}^1, \mathbf{X}^2, \cdots, \mathbf{X}^K]\begin{bmatrix}\hat{\mathbf{w}}^1 \\ \hat{\mathbf{w}}^2 \\ \vdots \\ \hat{\mathbf{w}}^K\end{bmatrix}, \quad (22)$$

where $\tilde{\mathbf{x}}^*$ is an estimate of the original image feature $\mathbf{x}^*$ and $\hat{\mathbf{w}} = [[\hat{\mathbf{w}}^1]^{\mathrm{T}}, [\hat{\mathbf{w}}^2]^{\mathrm{T}}, \cdots, [\hat{\mathbf{w}}^K]^{\mathrm{T}}]^{\mathrm{T}}$. Based on the assumption that the testing image feature is a linear combination of a few image features from its corresponding class, non-zero valued elements in $\hat{\mathbf{w}}$ should be only in $\hat{\mathbf{w}}^i$ if the testing image feature belongs to the class i. The BCS approach [17, 20] assigns the testing image feature to the class i if it has the highest norm-2 magnitude of $\hat{\mathbf{w}}^i$.

However, when there are training image features with no or a very small number of points of interest, most of the resulting feature vectors are zero valued. This would allow large weight values in $\hat{\mathbf{w}}$ without detrimentally affecting the likelihood value when evaluating equation (6). These inappropriately large weight values can lead to a data being misclassified when using the l_2-norm of the weights as a classification mechanism. To overcome this problem this work proposes a classification approach based on a backpropagation process as described below. Note that the backpropagation here is a reconstruction process, in which the weights are propagated back in order to reconstruct the input feature, this is different with the backpropagation process as in neural network.

The proposed approach reconstructs the testing image feature by a BCS process in which the image features are represented by equation (22). Similar to SRC, the weight vector $\hat{\mathbf{w}}$ is separated into K vectors with each vector keeping the value in its corresponding weight locations and setting the remaining values to zero:

$$\begin{aligned}\begin{bmatrix}\hat{\mathbf{w}}^1 \\ \hat{\mathbf{w}}^2 \\ \vdots \\ \hat{\mathbf{w}}^K\end{bmatrix} &= \begin{bmatrix}\hat{\mathbf{w}}^1 \\ \mathbf{0} \\ \vdots \\ \mathbf{0}\end{bmatrix} + \begin{bmatrix}\mathbf{0} \\ \hat{\mathbf{w}}^2 \\ \vdots \\ \mathbf{0}\end{bmatrix} + \cdots + \begin{bmatrix}\mathbf{0} \\ \mathbf{0} \\ \vdots \\ \hat{\mathbf{w}}^K\end{bmatrix}, \\ \hat{\mathbf{w}} &= \tilde{\mathbf{w}}^1 + \tilde{\mathbf{w}}^2 + \cdots + \tilde{\mathbf{w}}^K, \end{aligned} \quad (23)$$

where $\tilde{\mathbf{w}}^i \in \mathbb{R}^{N\times 1}$ and $i \in \{1, 2, \cdots, K\}$. Each $\tilde{\mathbf{w}}^i$ is used to reconstruct the testing image feature $\mathbf{x}_{cons}^i$ as follows:

$$\mathbf{x}_{cons}^i = \mathbf{X}\tilde{\mathbf{w}}^i. \quad (24)$$

The testing image feature $\mathbf{x}^*$ is assigned to a class corresponding to the most similar reconstructed image feature. More specifically, if the testing image feature recovered by $\tilde{\mathbf{w}}^i$ has the highest similarity with the original testing image feature $\mathbf{x}^*$, then this testing image feature can be classified into the i^{th} class. In order to compute the similarity between the image feature recovered by $\tilde{\mathbf{w}}^i$ and the original image feature $\mathbf{x}^*$, an error term is defined for each class:

$$Err(i) = ||\mathbf{x}^* - \mathbf{x}_{cons}^i||_2. \quad (25)$$

Then the testing image feature can be classified into the class which gives the minimum error. SRC, BCS and BBCS classifiers all need a dictionary composed by training data, hence they are naturally in-efficient for large datasets.

B. Column-based Subspace Sampling

Estimating the coefficients in equation (5) for BBCS can be time consuming when $\mathbf{X}$ is high dimensional. PCA can solve this problem by mapping the data into a lower dimensional data space. However, as the space has been altered, each entry can be difficult to interpret. The column-based subspace

sampling method can avoid these problems [24]. It selects the "best" subset of h columns from $\mathbf{X}$, where $h < N$.

Let $\mathbf{X}_k$ represent the "best" rank-k approximation to $\mathbf{X}$ by singular value decomposition. The output matrix $\mathbf{D} \in \mathbb{R}^{M \times h}$ consists of h columns from $\mathbf{X}$ such that the inequality in equation (26) is valid for a probability at least $1 - \delta$.

$$||\mathbf{X} - \mathbf{D}\mathbf{D}^{+}\mathbf{X}||_F \leqslant (1+\rho)||\mathbf{X} - \mathbf{X}_k||_F, \tag{26}$$

where $||\cdot||_F$ is the Frobenius norm, $\mathbf{D}^{+}$ is a Moore-Penrose generalized inverse of $\mathbf{D}$, ρ is an error parameter and δ is the failure probability.

Define a score for each column in "$\mathbf{X}$" in the following form:

$$\pi_j = \frac{1}{k}\sum_{\xi=1}^{k}(\mathbf{v}_j^{\xi})^2, \tag{27}$$

where $\mathbf{v}_j^{\xi}$ (j=1,2,...,N) is the j^{th} coordinate of $\mathbf{v}^{\xi}$ and $\mathbf{v}^{\xi} \in \mathbb{R}^{N\times 1}$ ($\xi = 1, 2, \cdots, k$) is the top right k singular vectors of $\mathbf{X}$. A random sampling process is applied on $\mathbf{X}$ and the j^{th} column of $\mathbf{X}$ is adopted with probability $min\{1, h\pi_j\}$, where $h = O(k\log k/\rho^2)$. All the adopted columns then generate the target matrix $\mathbf{D}$, with h examples to represent the original dataset. The detailed proof is given in [24, 26].

V. Performance Evaluation for Vehicle Logo Recognition

The proposed BBCS can be used a generic classifier. In this paper, we implemented for Vehicle Logo Recognition (VLR) and Traffic Scene Recognition (TSR). Recognizing vehicle logos and traffic scenes is of paramount importance for intelligent transportation systems, especially for traffic monitoring and management. The vehicle logo is one of its most distinguishable vehicle features [11] and as part of systems it can facilitate detecting fraudulent plates even when the observed logo is not available in the police security database [35]. As a result, this could give robust vehicle identification also in commercial investigations [1] and documents retrieval systems [36]. VLR is also linked with TSR which plays a crucial role in self-driving cars, traffic safety [37] and surveillance [38].

In this section the open VLR dataset provided by Huang et al. [6] is used to evaluate the proposed classification approach. It has 10 categories and each category contains 1000 training images and 150 testing images. All images have a size of 70×70 pixels. Figure 1 shows an example of the 10 vehicle categories by randomly choosing one image from each category in the training dataset.

The local descriptor Scale Invariant Feature Transform (SIFT) [27] and the bag of words [39] model are applied in order to represent images before the classification. All SIFT interest points are clustered in order to generate a dictionary with M words. In the representation stage, interest points from an image are replaced by their nearest words in the dictionary. This allows each image to be represented as a feature vector of length M, where M is the number of centroids in the clustering process in the bag of words model. The value in each entry of the vector is the normalized frequency of each word which appeared in an image. Increasing M gives more detailed information about the feature but increases the computation costs. Further details about representation models can be found in [40, 41].

Fig. 1: Vehicle logo dataset.

The performance evaluation is conducted in MATLAB on a computer with the following specification: Intel CPU I5-4590 (3.4Ghz) and 8GB of RAM. The open source library VLFeat [42] is applied for extracting the SIFT features. A comparison is made with the SRC (implemented using CVX [43, 44]), BCS classifier and KNN classifier. In our experiment, K=1 achieves the best result for clear images. Different K values influence the result when images are noisy, while the prior knowledge of images is unknown. Therefore, as it is commonly done in the literature, [14, 25], here a value of $K = 1$ is selected for all considered examples. The performance of each method is evaluated in terms of accuracy (percentage of correctly classified images), the total number of misclassified images and the computation time (to indicate the relative computational complexities).

A. Classification comparisons for Vehicle Logo Recognition

This subsection compares the performances of the classification methods when applied to the images that are provided in the dataset [6]. The simulation is repeated 30 times, the average accuracy is found and given with the corresponding standard deviation. The computation time and number of misclassified images are also given as the mean results for all the simulation runs.

Table I shows that the BBCS classifier achieves the highest accuracy of 98.91%. Table I also indicates that the BCS classifier is less accurate than the SRC and BBCS classifier. For example, when M=300, the BCS classifier incorrectly classifies 138 images, while this is reduced to 17 images for the BBCS

TABLE I: Non-parametric classifiers comparison using SIFT descriptors with M=100, 200, 300, 400, 500.

Classifiers	K**NN**	**SRC**	**BCS**	**BBCS**
M=100 (Accuracy%)	98.29 ± 0.36	**98.30** ± 0.44	92.17 ± 0.77	98.24 ± 0.32
Misclassified images	25.65	**25.50**	117.45	26.40
Time (s)	**0.97**	6357	868	868
M=200 (Accuracy%)	98.72 ± 0.24	**98.73** ± 0.25	91.36 ± 0.54	98.60 ± 0.28
Misclassified images	19.20	**19.05**	129.60	21
Time (s)	**1.84**	7804	2358	2358
M=300 (Accuracy%)	98.63 ± 0.27	98.78 ± 0.24	90.77 ± 0.75	**98.86** ± 0.22
Misclassified images	20.55	18.30	138.45	**17.10**
Time (s)	**2.70**	8360	3120	3120
M=400 (Accuracy%)	98.67 ± 0.30	98.83 ± 0.23	90.37 ± 0.77	**98.91** ± 0.24
Misclassified images	19.95	17.55	144.45	**16.35**
Time (s)	**3.54**	9116	3360	3360
M=500 (Accuracy%)	98.74 ± 0.23	**98.86** ± 0.19	90.25 ± 0.95	98.84 ± 0.25
Misclassified images	18.90	**17.10**	146.25	17.40
Time (s)	**4.17**	9582	3497	3497

classifier. In this case, the number of misclassifications is reduced by 88% without increasing the computational cost. For all the values of M considered, there was a mean reduction in the number of misclassified logos of 87% for BBCS classifier as compared to the BCS classifier. The computation times in Table I show that this improvement in classification accuracy comes without an increase in computational complexity.

The SRC and BBCS classifier give very similar classification accuracies. However, the BBCS classifier has a significant advantage in terms of computational costs. For the example when M=300, the proposed BBCS classifier reduces the computational cost by 63% when compared with the SRC whilst giving a slightly improved accuracy compared with the SRC algorithm. When comparing the computation times of the proposed BBCS classifier to the SRC, for all values of M considered, there is a mean reduction in the computation time of 68%. It only takes about two seconds to recognize an image using the BBCS classifier (note, that the times in Table I are for classifying all images in the testing dataset). The computation times show that the KNN classifier is quicker than the proposed BBCS classification approach. However, later results will show that the KNN classifier is more vulnerable to the effects of noise than the BBCS approach.

According to these results the computation times for the BCS and BBSCS are the same. However, the accuracy is consistently lower for the BCS classifier as compared to the BBCS classifier. The accuracy of the other two classifiers considered in the comparison also outperforms the BCS based method. As a result, the BCS based classifier will not be considered further in this performance evaluation.

Figure 2 shows 20 images (from the original testing dataset) that the KNN algorithm fails to satisfactorily classify. The first row gives the images that are under consideration and the second row gives the classification results from the KNN classifier. For comparison the SRC and BBCS classification results are shown in rows 3 and 4, respectively. The relative performances of the three methods are also further summarized in Table II. Here it can be seen that both methods outperform the KNN algorithm in terms of classification accuracy. The BBCS classifier gives the highest classification accuracy overall. Note that the 30 independent simulation runs are conducted with the final selected class being the most frequent overall.

TABLE II: Accuracies obtained using challenging data.

Classifier	K**NN**	**SRC**	**BBCS**
Accuracy	19.17%	43.83%	**47**%

B. Classification comparisons with noise

In practice, it is unlikely that the logos being classified will be clearly visible. Hence, here different levels of Gaussian white noise are added to the training images and testing images in order to examine the performance of the classifiers. Due to computational costs only M=300 will be considered in this subsection and those that follow. This has been selected as a compromise between accuracy and computational costs.

Figure 3 shows an example of a training image and the effects of adding noise with increasing values of variance. The intensity of all pixels in the image are normalized, giving values between 0 and 1. A white Gaussian noise is then added to each pixel which varies the pixel intensity, with the effects of different variance levels being investigated. Normally an image is considered highly contaminated if the variance of the Gaussian noise is above 0.2. The noise variance level in the training and testing images are denoted as σ^2_{train} and σ^2_{test}, respectively.

Ten independent classification simulation runs are then carried out using the noisy images and the mean accuracies are shown in Figure 4. Although adding a small amount of noise to the training images can initially offer an improvement in terms of classification accuracy, there is a degradation in performance when it is is increased further.

According to the authors' experience, there are more SIFT features that could be detected in slightly noisy images. This results in a better image representation vector. It can be explained by the fact that the use of the small amount of noise preserves more edges than for clear images after the Gaussian smoothing process used in the SIFT algorithm. However, an increase of the noise level makes difficult to recover the image. As the noise variance is increased, less and less SIFT features can then be detected as the images are then severely damaged by the noise.

Figure 4 shows that the KNN classifier is the most vulnerable to the effects of noise. It can be explained by the fact that the KNN classifier only calculates the Euclidean distance, while the other two allow for some error when modeling a testing image feature as a linear combination of the training image features. The performances of the BBCS classifier and

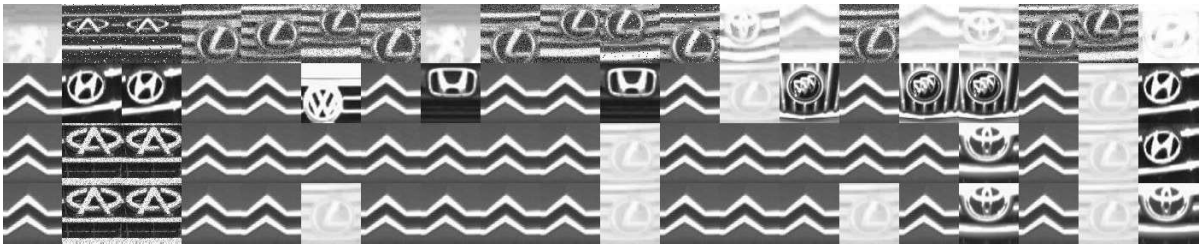

Fig. 2: The first row illustrates some challenge images, the second, third and fourth rows are the corresponding results classified by KNN, SRC and BBCS, respectively.

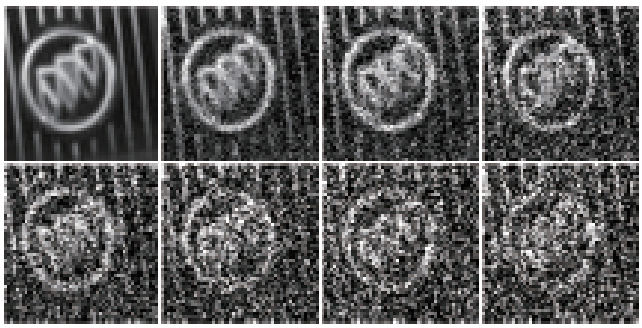

Fig. 3: An example of a training image and the effect by adding Gaussian white noise to image intensities with zero mean and variance values 0.02, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3 from left to right respectively.

the SRC are similar, while the BBCS classifier tends to be more accurate compared with the SRC when the training images are heavily contaminated by noise. For instance, when the noise variances are 0.25 in the training and testing images, the BBCS classifier and the SRC achieve 75.87 % and 73.79%, respectively. Furthermore, when the noise variances increase to 0.3, the BBCS classifier and the SRC can achieve 70.05 % and 67.82%, respectively.

C. Column-based subspace sampling

In this subsection, a reduced number of training images are used to evaluate the situation where the size of the dictionary is large. Table III shows the time and computational cost comparisons for different classifiers. Using the column-based subspace sampling method, the partial dictionary size is decreased to 20% and 10% (denoted as $p1$ and $p2$, respectively) when compared to the original dataset (denoted as f). The computational cost decreases about 6 times ($p1$) and 11 times ($p2$), while the accuracy drops slightly. The proposed BBCS approach requires an overall time 500 and 277 seconds, respectively, which is 0.3s and 0.18s per image.

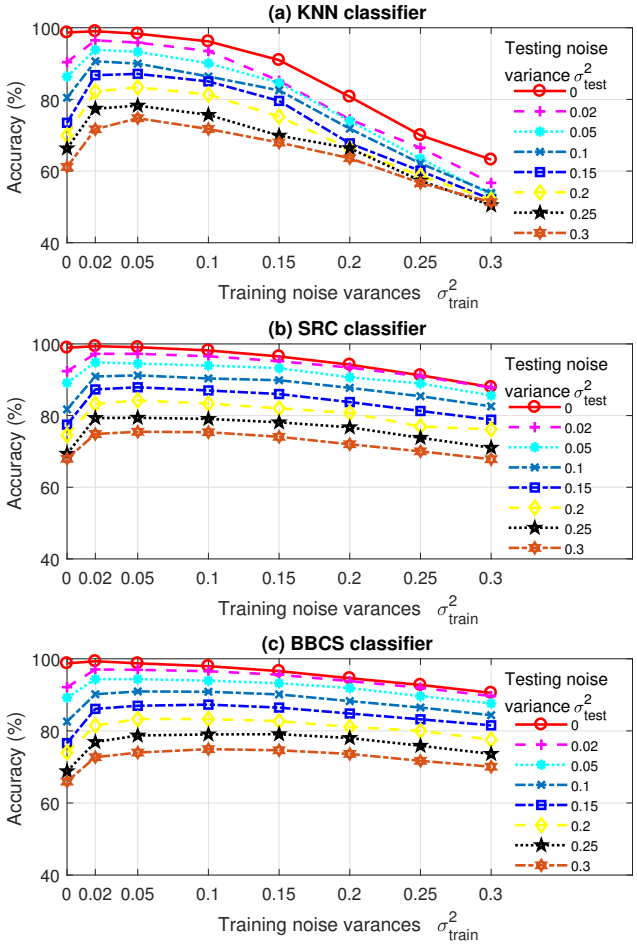

Fig. 4: Noise robustness comparisons for the KNN, SRC, BBCS classifiers.

The experiments are performed over 1500 images. This could still be applied to real-time applications. Even though the computational cost of the proposed algorithm is still higher than the cost of the KNN algorithm, it is more robust than the KNN when applied to noisy images. Since 10% data reduction does not decrease the accuracy significantly, the next

TABLE III: Comparisons between using the full and partial dictionaries.

Classifiers	K**NN**(f)	**SRC**(f)	**BBCS**(f)
Accuracy(%)	98.63 ± 0.27	98.78 ± 0.24	**98.86** ± 0.22
Misclassified images	26.33	18.30	**17.10**
Time(s)	**2.70**	8360	3120
Classifiers	K**NN**$(p1)$	**SRC**$(p1)$	**BBCS**$(p1)$
Accuracy(%)	97.32 ± 0.47	**98.54** ± 0.31	98.24 ± 0.35
Misclassified images	40.20	**21.83**	26.83
Time(s)	**0.25**	1436	500
Classifiers	K**NN**$(p2)$	**SRC**$(p2)$	**BBCS**$(p2)$
Accuracy(%)	96.75 ± 0.86	**97.49** ± 0.61	96.94 ± 0.52
Misclassified images	40.20	**21.83**	26.83
Time(s)	**0.13**	1170	277

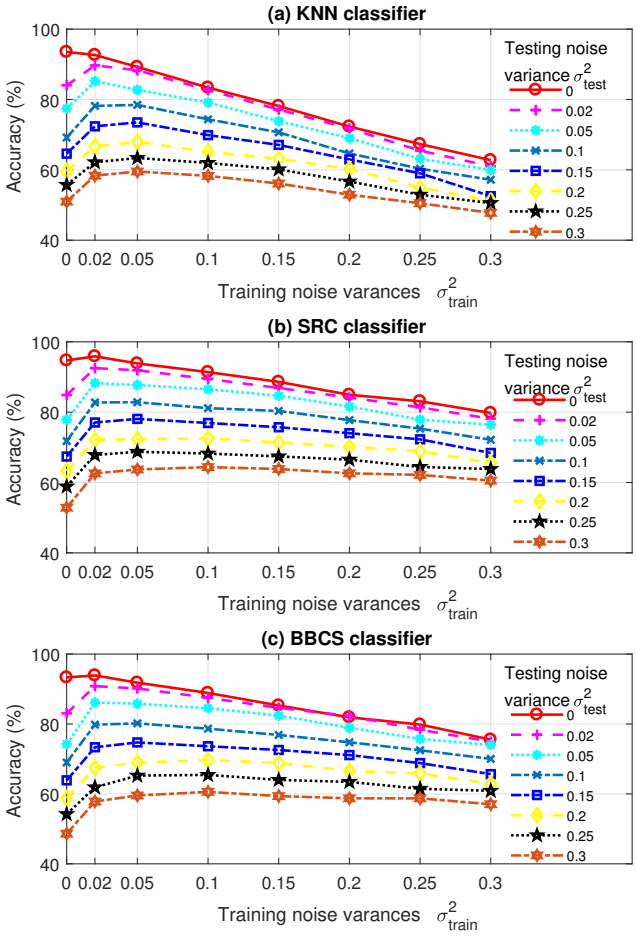

Fig. 5: Noise robustness comparisons when there are 10% training examples in each class using the column-based subspace sampling.

experiments are performed with 10% data reduction as a trade-off between the computational cost and accuracy.

Figure 5 shows the result of different classifiers when the dictionary size is decreased to 10% by the column-based subspace sampling method. When comparing the accuracies to those shown in Figure 4, the accuracy of each classification

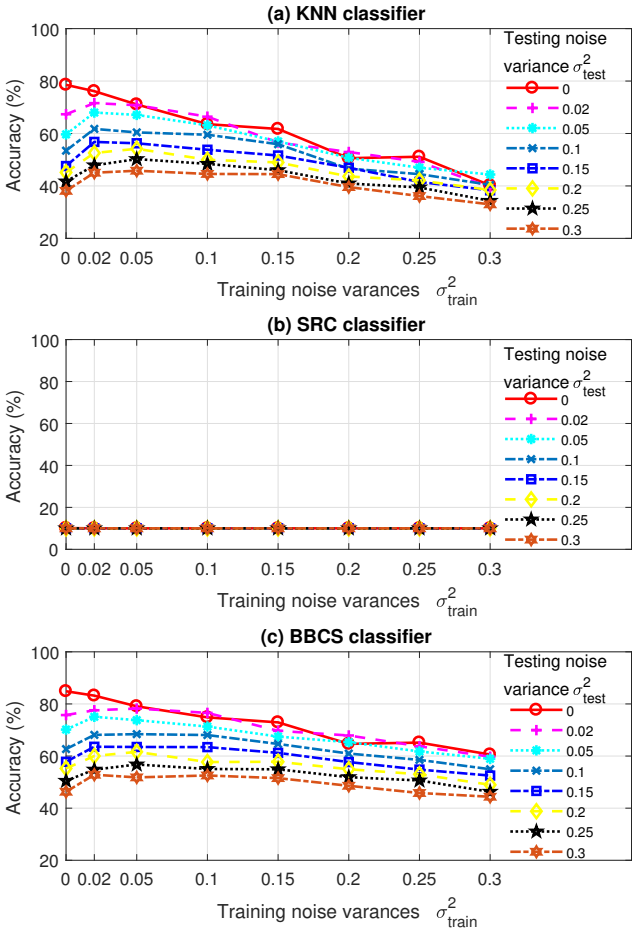

Fig. 6: Noise robustness comparisons when there are 1% training examples in each class using the column-based subspace sampling.

method has been reduced when compared to Figure 5. Moreover, Figure 5 shows that the KNN classifier is vulnerable to noise and the SRC is only marginally more accurate than the BBCS classifier, despite having previously been shown to be less computationally efficient. However, the computational cost is dropped as the dictionary size has decreased by 10 times.

The size of the training dataset is further decreased to only 1% selected images for each class in each of the 10 independent simulations, with the resulting classification accuracies being shown in Figure 6. In this case, the accuracies of the KNN classifier is not as high as the BBCS algorithm, especially when the noise levels increase. The SRC does not work any more since $M > N$ and the system is no longer under-determined. Note, that the conventional compressive sensing framework (as used in the SRC) is specifically designed for systems which are under-determined [15]. This leads to a random guess which can only achieve 10% accuracy as there are 10 classes with equal number of logos in each class.

Fig. 7: Example of classes from the FM2 dataset.

VI. Performance Evaluation for Scene Recognition

So far this section has considered the application of BBCS for VLR. Traffic scene recognition is a very similar topic in smart cities. Here the FM2 dataset [45] is considered. This dataset contains 6237 images from eight classes: highway, road, tunnel, tunnel exit, settlement, overpass, toll booth and dense traffic. Seventy percent of the images are randomly chosen for the training stage and the rest 30% of images are for testing purposes. Figure 7 illustrates some examples of the FM2 dataset. A pre-trained CNN framework (AlexNet [28]) is used for feature extraction. Instead of using the original weights from the network which was trained on other images, this work replaces the last fully connected layer to 200 neurons and fine tunes the weights based on traffic scene images. Hence, each image is represented by a vector of length 200. Note that the focus is on the classification method rather than on the image feature extraction.

The column-based subspace sampling representation is applied to each training group. Since each class has an imbalanced training data, the experiment set a maximum number of 200 to each class. When a class has more than 200 training images, the column-based subspace sampling method is applied to this class. A comparison with a recently developed deep learning approach, the CNN from [28] is performed, where the weights are trained for classification. Note that in CNN the classification is applied directly without using column-based subspace sampling. Since the parameters are fixed based on the whole training dataset, there is no need of retraining a network using a much smaller dataset. However, the results for KNN, BBCS and SRC are achieved on the new dataset after the column based sub-sampling.

Table IV shows the result from each classifier. Zero-mean Gaussian noises with different noise variance are applied on these training images and testing images. Without adding any noise, the CNN achieves the highest accuracy. However, when increasing the noise, the CNN becomes fragile. Similar research shows that when changing the intensity of even a single pixel the classification result changes [46]. However, using the extracted features from CNN and applying them to other classifiers leads to better results. Increasing the noise level, the proposed BBCS achieves the best results. This is important as the real images are not always clear. Figure 8 illustrates how different noise levels influence an image.

TABLE IV: Classifiers accuracy comparisons using features extracted by CNN based on FM2 dataset.

Noise variance	**CNN**(%)	K**NN**(%)	**SRC**(%)	**BBCS**(%)
0	**87.70**	84.41	87.00	86.31
0.01	57.01	73.21	79.73	**79.89**
0.1	10.59	56.04	57.59	**64.39**
0.2	7.43	52.03	42.51	**54.33**

VII. Application of BBCS to alternative dataset

The proposed BBCS approach has the potential to be applied to other areas, not only to VLR and TSR. In the performance validation the CIFAR-10 dataset [47] is used. This dataset consists of 50000 training images and 10000 testing images. Here, a CNN similar to [28] is trained on the new dataset. The network contains 3 convolution layers with 48, 96 and 192 3-by-3 kernels. Each convoluational layer is followed by a batch normalization layer and a max-pooling layer. Two fully connected layers are followed with 512 and

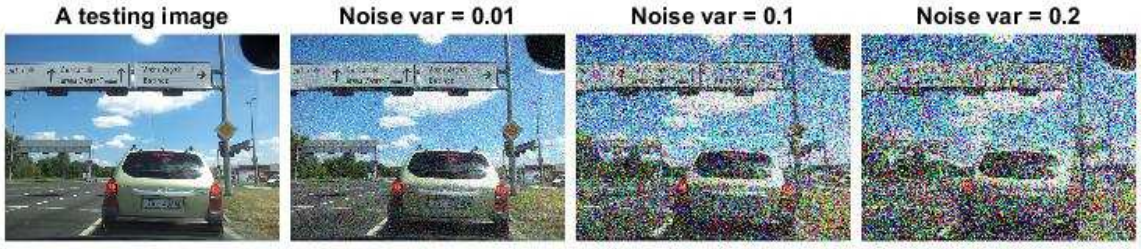

Fig. 8: An example of an traffic scene image with different level of noises.

200 neurons, respectively. The ReLU non-linear function [28] is applied to all neurons except the softmax being applied to the neurons in the last layer. The last fully connected layers is use as the feature. Hence, each image is represented by a vector of length 200.

The column-based subspace sampling is applied to each training group. This process picks 200 image feature vectors from 5000 image feature vectors in each group (4% of the original size). Hence, in order to avoid using all image feature vectors, the dictionary $\mathbf{X}$ is formed by only 2000 representative image feature vectors. Both the CNN and BBCS approaches train the weights for classification. Similarly, in CNN the classification is applied directly without using column-based subspace sampling.

TABLE V: Classifiers accuracy comparisons using features extracted by CNN based on CIFAR-10 dataset.

Noise variance	**CNN**(%)	K**NN**(%)	**SRC**(%)	**BBCS**(%)
0	**81.87**	68.79	78.53	73.40
0.01	47.60	52.77	52.51	**58.36**
0.02	36.37	42.39	43.80	**46.98**

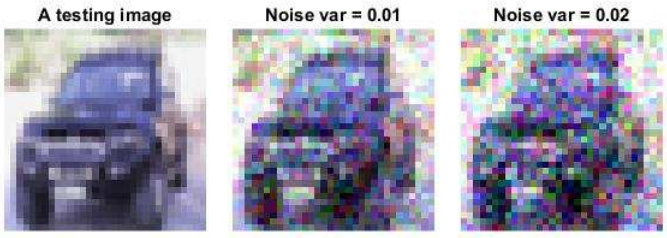

Fig. 9: An example of an image from CIFAR 10 with different level of noises.

Table V gives the performance of each classifier. Zero-mean Gaussian noises with different noise variance are added on these training images and testing images. Note that here the noise level is lower than the VLR dataset. The reason for this is the images in CIFAR-10 are tiny color images. A small color image can be easily contained by adding up the noise effects from each channel. Figure 9 illustrates the effect of the noise contamination. Similar to the TSR dataset, the result shows that the CNN classifier is not robust to noise. However, using the features extracted by the CNN and applying it to other classifiers could achieve better accuracy. This is important as clear images are not always guaranteed in real applications. Table V also shows that SRC should achieve good accuracy when the images are noise free, even if only 4% training images is applied. However, when the images are noisy, the BBCS algorithm achieves the best accuracy. Again, both BBCS and SRC perform better than the KNN algorithm.

VIII. Conclusion

This paper proposes a novel non-parametric classification approach, namely the BBCS classifier. The novelty of the work has two main components: *i*) the proposed back propagation process, *ii*) the proposed column-based subspace sampling to reduce the size of the dataset and associated computation costs. The developed approach relies on the constructing of the testing image feature using partial information from the weights estimated by BCS. Note, that for each class there is a corresponding reconstructed image feature. By comparing the reconstructed image feature with the testing image feature, the objects of interest are reconstructed and classified. The proposed backpropagation process gives a significant reduction of the misclassification error. For VLR, the number of misclassified testing images reduces by 87% when compared with the BCS classifier. Compared with the SRC, the BBCS algorithm gives a similar recognition accuracy while decreasing the mean computational cost by 68%. However, the SRC does not work when the training dataset is small while the BBCS algorithm shows accurate results in the same situation. Moreover, the proposed classifier and column based subspace sampling have been shown to be robust to the effects of heavy noise, unlike the KNN classifier. The proposed approach is a general non-parametric classifier and is also validated on TSR dataset and on the CIFAR-10 image dataset.

Acknowledgment

We appreciate the support of the "SETA project: An open, sustainable, ubiquitous data and service for efficient, effective, safe, resilient mobility in metropolitan areas" funded from

the European Union's Horizon 2020 research and innovation programme under grant agreement no. 688082.

IX. Appendix

A. Marginal likelihood maximization

The following gives a detailed derivation for the marginal likelihood in equation (17). By combining equations (6) and (7), the marginal likelihood can be expanded to:

$$\begin{aligned} p(\mathbf{x}^*|\boldsymbol{\alpha},\sigma^2) &= \int p(\mathbf{x}^*|\mathbf{w},\sigma^2)\ p(\mathbf{w}|\boldsymbol{\alpha})d\mathbf{w} \\ &= (2\pi\sigma^2)^{-\frac{M}{2}}(2\pi)^{-\frac{N}{2}}|\mathbf{A}|^{\frac{1}{2}} \\ &\times \int exp\left\{-\frac{1}{2\sigma^2}||\mathbf{x}^*-\mathbf{X}\mathbf{w}||_2^2+\mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{w}\right\}d\mathbf{w}. \quad (28) \end{aligned}$$

In order to simply equation (28), define:

$$\mathbb{Q} = \frac{1}{2}\left\{\frac{1}{\sigma^2}||\mathbf{x}^*-\mathbf{X}\mathbf{w}||_2^2+\mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{w}\right\}. \quad (29)$$

Combining with equations (13) and (14), (29) can be given as:

$$\mathbb{Q} = \frac{1}{2}\left(\frac{\mathbf{x}^{*\mathrm{T}}\mathbf{x}^*}{\sigma^2}-\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right)+\frac{1}{2}(\mathbf{w}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{w}-\boldsymbol{\mu}). \quad (30)$$

In order to simply equation (30), we set

$$\mathbb{T} = \frac{1}{2}\left(\frac{\mathbf{x}^{*\mathrm{T}}\mathbf{x}^*}{\sigma^2}-\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right), \quad (31)$$

Therefore the integral part in the right hand side of equation (28) is given by:

$$\int exp\{-\mathbb{Q}\}d\mathbf{w} = (2\pi)^{\frac{N}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}exp\{-\mathbb{T}\}. \quad (32)$$

Substituting (32) back in equation (28) gives:

$$p(\mathbf{x}^*|\boldsymbol{\alpha},\sigma^2) = (2\pi\sigma^2)^{-\frac{M}{2}}|\mathbf{A}|^{\frac{1}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}exp\{-\mathbb{T}\}. \quad (33)$$

This can be further simplified by:

$$p(\mathbf{x}^*|\boldsymbol{\alpha},\sigma^2) = (2\pi)^{-\frac{M}{2}}\frac{1}{\sigma^M}\frac{1}{|\mathbf{I}_N+\sigma^{-2}\mathbf{A}^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X}|^{\frac{1}{2}}}exp\{-\mathbb{T}\}, \quad (34)$$

where $\mathbf{I}_N = \mathbf{A}^{-1}\mathbf{A}$. Using the matrix determinants properties [48] that $|\mathbf{I}_N+\mathbf{D}^{\mathrm{T}}\mathbf{B}| = |\mathbf{I}_M+\mathbf{D}\mathbf{B}^{\mathrm{T}}|$ with $\mathbf{D}\in\mathbb{R}^{M\times N}$ and $\mathbf{B}\in\mathbb{R}^{M\times N}$, the above equation can be updated to:

$$p(\mathbf{x}^*|\boldsymbol{\alpha},\sigma^2) = (2\pi)^{-\frac{M}{2}}\frac{1}{|\sigma^2\mathbf{I}_M+\mathbf{X}\mathbf{A}^{-1}\mathbf{X}^{\mathrm{T}}|^{\frac{1}{2}}}exp\{-\mathbb{T}\}. \quad (35)$$

Recall the $\mathbb{T}$ is given in equation (31) and it can be expressed as follows:

$$\mathbb{T} = \tfrac{1}{2}\left(\mathbf{x}^{*\mathrm{T}}\left[\sigma^{-2}\mathbf{I}_M-\sigma^{-2}\mathbf{X}(\mathbf{A}+\sigma^{-2}\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\sigma^{-2}\right]\mathbf{x}^*\right). \quad (36)$$

According the Woodbury inversion identity [34]:

$$[\sigma^{-2}\mathbf{I}_M-\sigma^{-2}\mathbf{X}(\mathbf{A}+\sigma^{-2}\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\sigma^{-2}] = (\sigma^2\mathbf{I}_M+\mathbf{X}\mathbf{A}^{-1}\mathbf{X}^{\mathrm{T}})^{-1}, \quad (37)$$

equation (36) can be expressed as:

$$\mathbb{T} = \frac{1}{2}\left(\mathbf{x}^{*\mathrm{T}}(\sigma^2\mathbf{I}_M+\mathbf{X}\mathbf{A}^{-1}\mathbf{X}^{\mathrm{T}})^{-1}\mathbf{x}^*\right). \quad (38)$$

Therefore, equation (35) can be given as:

$$p(\mathbf{x}^*|\boldsymbol{\alpha},\sigma^2) = \frac{1}{\sqrt{(2\pi)^M|\mathbf{C}|}}exp\left\{-\frac{1}{2}\mathbf{x}^{*T}\mathbf{C}^{-1}\mathbf{x}^*\right\}, \quad (39)$$

which links back to equation (18), with the $M\times M$ matrix $\mathbf{C}$ is given by:

$$\mathbf{C} = \sigma^2\mathbf{I}_M+\mathbf{X}\mathbf{A}^{-1}\mathbf{X}^{\mathrm{T}}. \quad (40)$$

B. Evidence approximation

This subsection presents the derivation of the marginal log-likelihood function and its maximization with respect to α_i and σ^2. We can express $\mathbb{T}$ from equation (31) as follows:

$$\mathbb{T} = \frac{1}{2\sigma^2}||\mathbf{x}^*-\mathbf{X}\boldsymbol{\mu})||_2^2+\frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\mathbf{A}\boldsymbol{\mu}. \quad (41)$$

Hence, taking the logarithm of the marginal likelihood given in equation (33), the logarithm of the marginal likelihood can be obtained in the following form:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\alpha},\sigma^2) = &-\frac{M}{2}ln\sigma^2-\frac{M}{2}ln(2\pi)+\frac{1}{2}\sum_{i=1}^{N}ln\alpha_i \\ &+\frac{1}{2}ln|\boldsymbol{\Sigma}|-\frac{1}{2\sigma^2}||\mathbf{x}^*-\mathbf{X}\boldsymbol{\mu})||_2^2-\frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\mathbf{A}\boldsymbol{\mu}. \quad (42) \end{aligned}$$

The procedure of maximizing equation (42) with respect to α_i and σ^2 is known as the evidence approximation procedure.

Following the approach from [49], the derivative of $ln|\boldsymbol{\Sigma}|$ with respect to α_i is:

$$\frac{d}{d\alpha_i}ln|\boldsymbol{\Sigma}| = \frac{d}{d\alpha_i}-ln|\boldsymbol{\Sigma}|^{-1} = -\text{Trace}\boldsymbol{\Sigma} = -\Sigma_{ii}, \quad (43)$$

where Σ_{ii} is the i^{th} diagonal component of the posterior covariance matrix $\boldsymbol{\Sigma}$ and $Trace$ is the trace of a matrix. Therefore, the derivative of $\mathcal{L}(\boldsymbol{\alpha},\sigma^2)$ from equation (42) with respect to α_i is:

$$\frac{d\mathcal{L}(\boldsymbol{\alpha},\sigma^2)}{d\alpha_i} = \frac{1}{2\alpha_i}-\frac{1}{2}\Sigma_{ii}-\frac{1}{2}\mu_i^2. \quad (44)$$

Setting the derivative to zero, gives equation (20).

In order to simplify the $\frac{d\mathcal{L}(\boldsymbol{\alpha},\sigma^2)}{d\sigma^2}$, set $\beta = 1/\sigma^2$. Following the approach from [50], the derivative of $ln|\boldsymbol{\Sigma}|$ with respect to β is:

$$\begin{aligned} \frac{d}{d\beta}ln|\boldsymbol{\Sigma}| &= \frac{d}{d\beta}-ln|\boldsymbol{\Sigma}|^{-1} \\ &= -\text{Trace}(\mathbf{I}_N-\boldsymbol{\Sigma}\mathbf{A})\beta^{-1} \quad (45) \end{aligned}$$

Therefore, the derivative of $\mathcal{L}(\boldsymbol{\alpha}, \sigma^2)$ from equation (42) with respect to β is:

$$\frac{d\mathcal{L}(\boldsymbol{\alpha}, \sigma^2)}{d\beta} = \frac{M}{2\beta} - \frac{1}{2}||\mathbf{x}^* - \mathbf{X}\boldsymbol{\mu})||_2^2 - \frac{1}{2}Trace(\mathbf{I}_N - \boldsymbol{\Sigma}\mathbf{A})\beta^{-1}. \tag{46}$$

Setting the derivative to zero gives equation (21).

REFERENCES

[1] Y. Ou, H. Zheng, S. Chen, and J. Chen, "Vehicle logo recognition based on a weighted spatial pyramid framework," in *Proc. of 17th IEEE International Conf. on Intelligent Transportation Systems*, Qingdao, China, Oct. 2014, pp. 1238–1244.

[2] Q. Sun, X. Lu, L. Chen, and H. Hu, "An improved vehicle logo recognition method for road surveillance images," in *Proc. of 7th International Symposium on Computational Intelligence and Design*, Hangzhou, China, Dec 2014, pp. 373–376.

[3] D. Llorca, R. Arroyo, and M. Sotelo, "Vehicle logo recognition in traffic images using HOG features and SVM," in *Proc. of 16th IEEE International Conf. on Intelligent Transportation Systems*, The Hague, Netherlands, Oct. 2013, pp. 2229–2234.

[4] I. Sikirić, K. Brkić, J. Krapac, and S. Šegvić, "Robust traffic scene recognition with a limited descriptor length," in *Proc. of CVPR Workshop on Visual Place Recognition in Changing Environments*, Boston, Massachusetts, USA, June 2015.

[5] R. Chen, M. Hawes, L. Mihaylova, J. Xiao, and W. Liu, "Vehicle logo recognition by spatial-SIFT combined with logistic regression," in *Proc. IEEE International Conf. on Information Fusion*, Heidelberg, Germany, July 2016, pp. 1228–1235.

[6] Y. Huang, R. Wu, Y. Sun, W. Wang, and X. Ding, "Vehicle logo recognition system based on convolutional neural networks with a pretraining strategy," *IEEE Trans. on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1951–1960, 2015.

[7] R. Chen, M. A. Jalal, L. Mihaylova, and R. K. Moore, "Learning capsules for vehicle logo recognition," in *Proceedings of the 21st International Conference on Information Fusion*, July 2018, pp. 565–572.

[8] J. S. Sánchez, F. Pla, and F. J. Ferri, "On the use of neighbourhood-based non-parametric classifiers," *Pattern Recognition Letters*, vol. 18, no. 11, pp. 1179–1186, 1997.

[9] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, USA: Springer, 2006.

[10] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: The MIT Press, 2012.

[11] A. P. Psyllos, C.-N. E. Anagnostopoulos, and E. Kayafas, "Vehicle logo recognition using a SIFT-based enhanced matching scheme," *IEEE Trans. on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 322–328, 2010.

[12] H. Peng, X. Wang, H. Wang, and W. Yang, "Recognition of low-resolution logos in vehicle images based on statistical random sparse distribution," *IEEE Trans. on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 681–691, 2015.

[13] A. Hinneburg, C. C. Aggarwal, and D. A. Keim, "What is the nearest neighbor in high dimensional spaces?" in *Proc. of the International Conf. on Very Large Data Bases*, San Francisco, CA, USA, Sept. 2000, pp. 506–515.

[14] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[15] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[16] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. on Signal Processing*, vol. 56, no. 6, pp. 2346–2356, 2008.

[17] X. Zhang, J. Qin, and G. Li, "SAR target classification using Bayesian compressive sensing with scattering centers features," *Progress In Electromagnetics Research*, vol. 136, pp. 385–407, 2013.

[18] Y. Zhang, Y. Li, Z. Wang, Z. Song, R. Lin, J. Qian, and J. Yao, "A fast image reconstruction method based on bayesian compressed sensing for the undersampled AFM data with noise," *Measurement Science and Technology*, vol. 30, no. 2, p. 025402, jan 2019.

[19] Y. Huang, J. Paisley, Q. Lin, X. Ding, X. Fu, and X. Zhang, "Bayesian nonparametric dictionary learning for compressed sensing mri," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5007–5019, Dec 2014.

[20] T. N. Sainath, A. Carmi, D. Kanevsky, and B. Ramabhadran, "Bayesian compressive sensing for phonetic classification," in *Proc. of IEEE International Conf. on Acoustics Speech and Signal Processing*, Dallas, TX, USA, Mar 2010, pp. 4370–4373.

[21] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. of Advances in Neural Information Processing Systems*, Whistler, British Columbia, Canada, Dec. 2003, pp. 321–328.

[22] M. Shi, T. Furon, and H. Jégou, "A group testing framework for similarity search in high-dimensional spaces," in *Proc. of ACM International Conf. on Multimedia*, New York, NY, USA, Nov. 2014, pp. 407–416.

[23] A. Iscen, M. Rabbat, and T. Furon, "Efficient large-scale similarity search using matrix factorization," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, June 2016, pp. 2073–2081.

[24] M. W. Mahoney and P. Drineas, "CUR matrix decompositions for improved data analysis," *Proceedings of the National Academy of Sciences*, vol. 106, no. 3, pp. 697–702, 2009.

[25] E. Elhamifar, G. Sapiro, and R. Vidal, "See all by looking at a few: Sparse modeling for finding representative objects," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Providence, RI, USA, June 2012, pp. 1600–1607.

[26] C. Boutsidis, M. W. Mahoney, and P. Drineas, "An improved approximation algorithm for the column subset selection problem," in *Proc. of Annual ACM-SIAM Symposium on Discrete Algorithms*, New York, New York, Jan. 2009, pp. 968–977.

[27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. of 25th International Conf. on Neural Information Processing Systems*, Lake Tahoe, Nevada, USA, Dec. 2012, pp. 1097–1105.

[29] S. A. Tesfamicael and F. Barzideh, "Bayesian inference and compressed sensing," *Bayesian Inference, IntechOpen, 2017*, Nov 2017.

[30] E. Amaldi and V. Kann, "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems," *Theoretical Computer Science*, vol. 209, no. 1, pp. 237–260,

1998.

[31] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[32] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[33] D. L. Donoho, Y. Tsaig, I. Drori, and J. L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Trans. on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[34] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *The Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[35] L. Figueiredo, I. Jesus, J. T. Machado, J. Ferreira, and J. M. De Carvalho, "Towards the development of intelligent transportation systems," in *Proc. of IEEE Intelligent Transportation Systems*, Oakland, CA, USA, Aug. 2001, pp. 1206–1211.

[36] Z. Zhang, X. Wang, W. Anwar, and Z. L. Jiang, "A comparison of moments-based logo recognition methods," in *Proc. of Abstract and Applied Analysis*, vol. 2014, 2014, pp. 1–8.

[37] C. Y. Chen, W. Choi, and M. Chandraker, "Atomic scenes for scalable traffic scene recognition in monocular videos," in *Proc. of IEEE Winter Conf. on Applications of Computer Vision*, Lake Placid, NY, USA, Mar. 2016, pp. 1–9.

[38] T. Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S. J. Russell, and J. Weber, "Automatic symbolic traffic scene analysis using belief networks," in *Advancement of Artificial Intelligence*, vol. 94, 1994, pp. 966–972.

[39] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. of Workshop on Statistical Learning in Computer Vision*, Prague, Czech Republic, May 2004, pp. 1–22.

[40] X. Zhen and L. Shao, "Action recognition via spatio-temporal local features: A comprehensive study," *Image and Vision Computing*, vol. 50, pp. 1–13, 2016.

[41] U. L. Altintakan and A. Yazici, "Towards effective image classification using class-specific codebooks and distinctive local features," *IEEE Trans. on Multimedia*, vol. 17, no. 3, pp. 323–332, 2015.

[42] A. Vedaldi and B. Fulkerson, "VLFeat - an open and portable library of computer vision algorithms," in *Proc. of 18th ACM International Conf. on Multimedia*, New York, NY, USA, Oct. 2010, pp. 1469–1472.

[43] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar 2014.

[44] M. C. Grant and S. P. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent advances in learning and control*. London, UK: Springer, 2008, pp. 95–110.

[45] I. Sikiri, K. Brki, J. Krapac, and S. egvi, "Image representations on a budget: Traffic scene classification in a restricted bandwidth scenario," in *Proc. IEEE Intelligent Vehicles Symposium Proceedings*, Dearborn, MI, USA, June 2014, pp. 845–852.

[46] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *CoRR*, vol. abs/1710.08864, 2017.

[47] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Master's Thesis, Department of Computer Science, University of Toronto*, 2009.

[48] M. Brookes, "The matrix reference manual," *Imperial College London*, 2005. [Online]. Available: http://www.ee.imperial.ac.uk/hp/staff/dmb/matrix/intro.html

[49] D. J. MacKay, "Bayesian interpolation," *Neural computation*, vol. 4, no. 3, pp. 415–447, 1992.

[50] T. Fletcher, "Relevance vector machines explained," *University College London: London, UK*, 2010. [Online]. Available: http://home.mit.bme.hu/ horvath/IDA/RVM.pdf