SPARSE REPRESENTATIONS WITH CONE ATOMS

Denis C. Ilie-Ablachim, Andra Băltoiu, Bogdan Dumitrescu

University Politehnica of Bucharest
Department of Automatic Control and Computers
313 Spl. Independenţei, 060042 Bucharest, Romania

ABSTRACT

We extend the notion of sparse representation to the case where the atoms are not vectors, but cones, hence infinite sets. The sparse representation is linear, as usual, but the most convenient vector is chosen from each selected cone. We give a cone version of Orthogonal Matching Pursuit (OMP) and show that its complexity is only a few times larger than that of OMP. The new cone OMP can be used for anomaly detection; we apply it with very good results to the detection of abnormal heartbeats.

Index Terms— sparse representations, dictionary, Orthogonal Matching Pursuit, cone atoms, anomaly detection

1. INTRODUCTION

Sparse representations [1] are now used for solving many problems, like denoising, inpainting, compression, coding, compressed sensing, and also for machine learning tasks like classification. The standard sparse representation problem seeks to approximate a vector $\boldsymbol{y} \in \mathbb{R}^m$ with a linear combination of few columns of a given dictionary $\boldsymbol{D} \in \mathbb{R}^{m \times n}$, which can be chosen or trained. So, we want to minimize $\|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\|$, where $\boldsymbol{x} \in \mathbb{R}^n$ has only few nonzero elements. The columns of $\boldsymbol{D}$ are called atoms.

Problem. We propose here to extend the notion of atom from a single vector to an infinite set. As an interesting particular case, we propose the use of *cone atoms*. Denoting $\boldsymbol{d} \in \mathbb{R}^m$, with $\|\boldsymbol{d}\| = 1$, the central vector, the cone $\mathcal{C}(\boldsymbol{d}, \rho)$ contains all vectors $\boldsymbol{a} \in \mathbb{R}^m$, with $\|\boldsymbol{a}\| = 1$, for which $\|\boldsymbol{a} - \boldsymbol{d}\| \leq \rho$; we call ρ the radius of the cone. Note that $\mathcal{C}(\boldsymbol{d}, \rho)$ is stricto sensu a hypersector of the unit hypersphere in $\mathbb{R}^m$ rather than a cone; however, as a single atom is in fact used to describe a direction in the context of sparse representation, the set $\mathcal{C}(\boldsymbol{d}, \rho)$ extends to a (infinite) cone.

We interpret now the j-th column of the dictionary $\boldsymbol{D}$, denoted $\boldsymbol{d}_j$, as the central vector of a cone $\mathcal{C}(\boldsymbol{d}_j, \rho_j)$, with given radius ρ_j. So, besides the dictionary, we need only a set of n radii associated with the atoms. To build a sparse representation with cone atoms, we use any vector from a cone as an atom, with the same purpose as in the standard case, to minimize the representation error. The sparse representation problem becomes the optimization problem

$$\begin{array}{ll} \min\limits_{\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{a}_j \in \mathbb{R}^m} & \|\boldsymbol{y} - \sum_{j=1}^n \boldsymbol{a}_j x_j\|_2 \\ \text{s.t.} & \|\boldsymbol{x}\|_0 \leq s \\ & \boldsymbol{a}_j \in \mathcal{C}(\boldsymbol{d}_j, \rho_j),\ j = 1 : n \end{array} \tag{1}$$

Here, we name $\boldsymbol{a}_j$ actual atoms, since they are effectively used in the representation. The sparsity level is s. Figure 1 illustrates the solution of (1) for $s = 2$ cones. The optimal approximation of the signal $\boldsymbol{y}$ is its projection on the plane generated by $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$; these vectors are such that the plane is tangent to both cones, thus minimizing the distance to $\boldsymbol{y}$.

Although (1) looks much more difficult than the standard sparse representation problem, we will present here an efficient algorithm in the style of Orthogonal Matching Pursuit (OMP) [2]. Another apparent drawback of our model is the lack of the reconstruction capability. Indeed, after solving (1), the reconstruction is possible only if the actual atoms $\boldsymbol{a}_j$ are stored, since they are different for each signal. This may be impractical, compared with the standard case where only the indices of the atoms must be stored. So, applications in coding or compressed sensing are impossible. However, there are many important applications where reconstruction is not needed, like denoising, inpainting, classification or anomaly detection, where it is enough to store the approximation of $\boldsymbol{y}$ resulted from (1) or the representation error.

Relation to prior work. Our sparse representation method appears to be the first where the atoms can belong to an infinite set. Several vectors, represented by a single instance, are used in shift invariant dictionaries [3–5]. Also, multiple vectors can be represented with condensed information, in structured dictionaries like multi layer [6, 7], multi scale [8], separable [9].

Although (1) is linear for a signal, we can also say that there is intrinsic nonlinearity in it when multiple signals are involved, as it is typically the case, since the actual atoms $\boldsymbol{a}_j$ are not fixed, but chosen from an infinite set. Not only the standard sparse representation model is considerably en-

This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS - UEFISCDI, project number PN-III-P4-PCE-2021-0154, within PNCDI III.

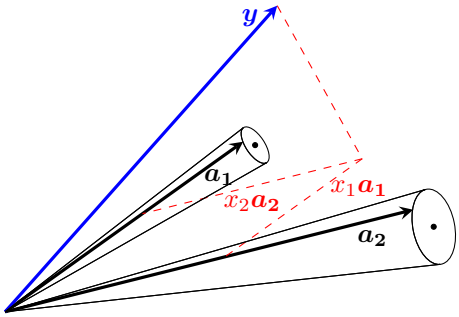

Fig. 1: Optimal approximation of a signal $\boldsymbol{y}$ with the linear combination $x_1\boldsymbol{a}_1 + x_2\boldsymbol{a}_2$ of two cone atoms.

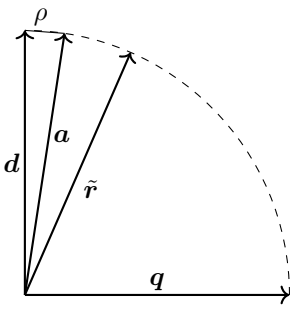

Fig. 2: Nearest atom $\boldsymbol{a}$ in cone centered in $\boldsymbol{d}$ with radius ρ, with respect to a vector $\tilde{\boldsymbol{r}}$. The vector $\boldsymbol{q}$ is orthogonal on $\boldsymbol{d}$. All vectors have length 1.

riched, but also further nonlinearity can be added, since kernel techniques [10] or their approximations [11] are applicable to our model. Related work on connections of dictionary learning with neural networks [12–14] is also of interest.

Contents of the paper. Section 2 presents the algorithm for computing the projection of an arbitrary vector on a cone or, equivalenty, for finding the nearest atom from that cone. This operation will be used in Section 3 to develop a cone version of OMP. Section 4 shows that our new Cone-OMP can be successfully used in anomaly detection, illustrating its use for the detection of abnormal heartbeats.

2. NEAREST CONE ATOM

The basic problem that appears in representation is the following. Given a vector $\tilde{\boldsymbol{r}} \in \mathbb{R}^m$, typically a residual, normalized such that $\|\tilde{\boldsymbol{r}}\| = 1$, we want to find the nearest atom from a given cone $\mathcal{C}(\boldsymbol{d}, \rho)$, i.e., the solution of

$$\begin{array}{ll} \min\limits_{\boldsymbol{a}\in\mathcal{C}(\boldsymbol{d},\rho)} & \|\boldsymbol{a} - \tilde{\boldsymbol{r}}\|_2 \\ \text{s.t.} & \|\boldsymbol{a}\| = 1 \end{array} \tag{2}$$

This is a simple geometric problem, illustrated in Fig. 2. If $\tilde{\boldsymbol{r}} \in \mathcal{C}(\boldsymbol{d}, \rho)$, then the trivial solution is $\boldsymbol{a} = \tilde{\boldsymbol{r}}$. Otherwise, it is clear that $\boldsymbol{a}$ lies in the subspace generated by $\boldsymbol{d}$ and $\tilde{\boldsymbol{r}}$ and is on the frontier of the cone, hence $\|\boldsymbol{a} - \boldsymbol{d}\| = \rho$. Let $\boldsymbol{q}$ be the vector orthogonal on $\boldsymbol{d}$ in this subspace, with $\|\boldsymbol{q}\| = 1$. We denote $p = \boldsymbol{d}^T\tilde{\boldsymbol{r}}$ and remark that we can consider $p > 0$ (we can replace $\tilde{\boldsymbol{r}}$ by $-\tilde{\boldsymbol{r}}$). We can write $\boldsymbol{q} = \alpha\boldsymbol{d} + \beta\tilde{\boldsymbol{r}}$. Orthogonality implies $0 = \boldsymbol{d}^T\boldsymbol{q} = \alpha + \beta p$. Using

$$\alpha = -\beta p \tag{3}$$

in $1 = \|\boldsymbol{q}\|^2 = \alpha^2 + \beta^2 + 2\alpha\beta p = \beta^2 p^2 + \beta^2 - 2\beta^2 p^2$, we get

$$\beta = 1/\sqrt{1 - p^2}. \tag{4}$$

We now have an orthogonal base for the 2-dimensional subspace where $\boldsymbol{a}$ lies and we write

$$\boldsymbol{a} = \lambda\boldsymbol{d} + \mu\boldsymbol{q}. \tag{5}$$

Algorithm 1: Nearest_atom: compute nearest atom in a cone from a given vector.

Data: vector $\boldsymbol{d} \in \mathbb{R}^m$ and $\rho > 0$ defining $\mathcal{C}(\boldsymbol{d}, \rho)$
vector $\boldsymbol{r} \in \mathbb{R}^m$
Result: atom $\boldsymbol{a} \in \mathcal{C}(\boldsymbol{d}, \rho)$ nearest from $\boldsymbol{r}$

1 Normalize vector: $\tilde{\boldsymbol{r}} = \boldsymbol{r}/\|\boldsymbol{r}\|$
2 Compute projection $p = \boldsymbol{d}^T\tilde{\boldsymbol{r}}$
3 **if** $p < 0$ **then**
4 Change orientation: $\tilde{\boldsymbol{r}} \leftarrow -\tilde{\boldsymbol{r}}$
5 **if** $|p| \geq 1 - \rho^2/2$ **then**
6 The vector is in the cone: $\boldsymbol{a} = \tilde{\boldsymbol{r}}$
7 **else**
8 Set $\beta = 1/\sqrt{1 - p^2}$, $\alpha = -\beta|p|$
9 Set $\lambda = 1 - \rho^2/2$, $\mu = \sqrt{1 - \lambda^2}$
10 The desired atom is $\boldsymbol{a} = (\lambda + \mu\alpha)\boldsymbol{d} + \mu\beta\tilde{\boldsymbol{r}}$

To find the coefficients of the linear combination we see that $\|\boldsymbol{a}\| = 1$ gives $\lambda^2 + \mu^2 = 1$ and $\|\boldsymbol{a} - \boldsymbol{d}\| = \rho$ gives

$$\boldsymbol{d}^T\boldsymbol{a} = 1 - \rho^2/2. \tag{6}$$

Multiplying (5) with $\boldsymbol{d}^T$ we get

$$\lambda = \boldsymbol{d}^T\boldsymbol{a} = 1 - \rho^2/2 \tag{7}$$

and take (μ is positive by construction)

$$\mu = \sqrt{1 - \lambda^2}. \tag{8}$$

Putting together all the relations above gives Algorithm 1. The vector $\boldsymbol{r}$ can be arbitrary. The condition that $\tilde{\boldsymbol{r}} \in \mathcal{C}(\boldsymbol{d}, \rho)$, used in step 5, results immediately from $\|\boldsymbol{d} - \tilde{\boldsymbol{r}}\| \leq \rho$, since the vectors have norm equal to 1; compare with (6).

3. OMP WITH CONE ATOMS

We proceed now to adapt OMP for solving (1). OMP has two important operations: the choice of the next atom in the sup-

port and the least squares calculation of the current solution. We examine their form for a dictionary with cone atoms.

Denoting $\boldsymbol{A}$ the matrix whose columns are the atoms selected before the current OMP step and $\boldsymbol{x}$ the associated representation vector, the current residual is $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}$. The next atom to enter the support is that with the largest projection on the residual (i.e. the nearest from the residual; see expression in the last line of Algorithm 1). To compute the best projection, we also use (3), (4), (8) and obtain

$$\begin{aligned} \tilde{\boldsymbol{r}}^T\boldsymbol{a} &= (\lambda + \alpha\mu)p + \beta\mu \\ &= \lambda p + \beta\mu(1-p^2) \\ &= \lambda p + \sqrt{(1-\lambda^2)(1-p^2)} \end{aligned} \tag{9}$$

We note that for computing this expression we do not need to compute the optimal atom $\boldsymbol{a}$, but only the standard OMP projection $p = \tilde{\boldsymbol{r}}^T\boldsymbol{d}$ (remind the sign convention making $p > 0$). Since λ is an atom-dependent constant that can be computed only once for a dictionary, it results that computing (9) for all atoms costs only slightly more than computing the projections in the OMP algorithm.

Once the support is increased with the atom with the best projection, OMP proceeds by computing the least squares solution for that support. This minimizes the error in the standard OMP setup. However, with cone atoms, this is no longer true. The atoms already selected by projection may be not optimal and they are usually not the solution of the current version of (1); see again Figure 1.

Since each cone is a convex set, the linear combination of atoms that optimally approximates $\boldsymbol{y}$ can be found by successive projections on the cones $\mathcal{C}(\boldsymbol{d}_j, \rho_j)$ that have been selected as support. Each projection is made using Algorithm 1. One can iterate the projections until there is no significant progress in the approximation error or for a predefined number of iterations. The natural initialization is the least squares approximation of $\boldsymbol{y}$ with the currently selected atoms. So, in each iteration, we compute a least squares solution, like in OMP, and perform several rounds of projections, whose additional complexity is relatively small, since only vectorial operations of $O(m)$ complexity are used in each run of Algorithm 1. The overall complexity is only a few times that of OMP.

The above operations are gathered in Algorithm 2, Cone-OMP. Steps 5-11 are the selection of the next cone atom and the computation of the atom that is nearest from the current residual. Step 7 implements (9) for next cone atom selection; elementwise multiplication is denoted by $\odot$. The matrix $\boldsymbol{A}$ contains the actual atoms from the selected cones. The for loop 14 implements a single round of projections, although several could be used by simply repeating the loop.

In this form of the algorithm, the representation vector $\boldsymbol{x}$ has finally s elements, corresponding to the s columns of the actually used dictionary $\boldsymbol{A}$. The cones to which the atoms belong are given by $\mathcal{S}$. We have ignored the case where the residual is inside a cone and hence the representation is exact; in this case, the algorithm must be stopped immediately.

Algorithm 2: Cone-OMP: Optimal Matching Pursuit with cone atoms using successive projections.

Data: dictionary $\boldsymbol{D} \in \mathbb{R}^{m\times n}$ and radii $\boldsymbol{\rho} \in \mathbb{R}^n$
($\boldsymbol{D}$ and $\boldsymbol{\rho}$ define cones $\mathcal{C}_j(\boldsymbol{d}_j, \rho_j)$, $j = 1:n$)
vector $\boldsymbol{y} \in \mathbb{R}^m$
sparsity level s
Result: representation $\boldsymbol{x} \in \mathbb{R}^s$, solution of (1)
support $\mathcal{S} \subset 1:n$ of the sparse representation
actually used dictionary $\boldsymbol{A} \in \mathbb{R}^{m\times s}$

1 Initialize $\boldsymbol{r} = \boldsymbol{y}$, $\mathcal{S} = \emptyset$, $\boldsymbol{A} = []$
2 **for** $j = 1$ **to** n **do**
3 $\lambda_j = 1 - \rho_j^2/2$
4 **for** $k = 1$ **to** s **do**
5 Normalized residual: $\tilde{\boldsymbol{r}} = \boldsymbol{r}/\|\boldsymbol{r}\|$
6 Compute projections: $\boldsymbol{p} = \boldsymbol{D}^T\boldsymbol{r}$
7 Decide next index: $j =$
 $\operatorname{argmax}\left(\boldsymbol{\lambda} \odot |\boldsymbol{p}| + \sqrt{(\boldsymbol{1} - \boldsymbol{\lambda}\odot\boldsymbol{\lambda}) \odot (\boldsymbol{1} - \boldsymbol{p}\odot\boldsymbol{p})}\right)$
8 Next atom: $\boldsymbol{a} = \text{Nearest\_atom}(\boldsymbol{d}_j, \rho_j, \tilde{\boldsymbol{r}})$
9 Increase support: $\mathcal{S} \leftarrow \mathcal{S} \cup \{j\}$
10 Update actual dictionary: $\boldsymbol{A} \leftarrow [\boldsymbol{A}\ \boldsymbol{a}]$
11 Compute $\boldsymbol{x}$ by solving $\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|$
12 Compute residual: $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}$
13 **if** $k > 1$ **then**
14 **for** $i = 1$ **to** k **do**
15 Index in full dictionary $j = \mathcal{S}(i)$
16 Remove current atom from residual:
 $\boldsymbol{r} \leftarrow \boldsymbol{r} + x_i\boldsymbol{a}_i$
17 Update atom:
 $\boldsymbol{a}_i = \text{Nearest\_atom}(\boldsymbol{d}_j, \rho_j, \boldsymbol{r})$
18 Update coefficient: $x_i = \boldsymbol{r}^T\boldsymbol{a}_i$
19 Update residual: $\boldsymbol{r} \leftarrow \boldsymbol{r} - x_i\boldsymbol{a}_i$

4. APPLICATION TO ANOMALY DETECTION

We apply Cone-OMP for detecting anomalies in MIT-BIH arrythmia database [15]. We analyze record #109, like in [16], using similar preprocessing. It is sampled at 360 Hz and covers about 30 minutes. It contains 2530 heartbeats, the anomalies being 38 premature ventricular contractions (PVC) and 2 fusion of ventricular and normal beats. We extract all windows of length 256 and we reduce the dimension of the signals to 32 via PCA. We divide the windows into 6 segments of about 5 minutes (108000 signals) each. We train a dictionary for each segment using 40 iterations of K-SVD [17], initialized with n atoms that are random linear combinations of the signals. Finally, for each window, we store the approximation errors for OMP and Cone-OMP.

For each heartbeat, we use the representation errors on 201 windows: the one centered on the R point, 100 to the left and 100 to the right. The median error is chosen as rep-

n	s	ρ	ROC AUC	FP TP=40	FP TP=39	FP TP=38
96	3	-	0.99971	10.3	6.4	4.2
96	4	-	0.99971	11.7	6	3.6
128	4	-	0.99968	10.2	6.7	4.9
64	3	0.08	0.99984	6.7	4.7	2.4
96	3	0.05	0.99980	6.3	4	2.8
64	4	0.07	0.99986	5.7	2.8	1.8
96	4	0.05	0.99989	6.7	3	0.9
128	4	0.05	0.99977	8.3	3.7	2.8

Table 1: Best results for OMP (upper part) and Cone-OMP (lower part).

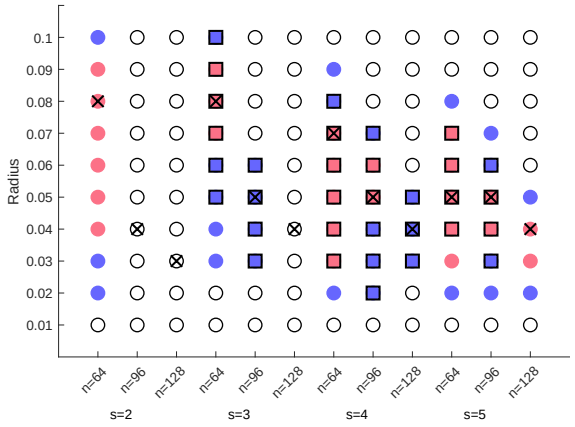

Fig. 3: Comparison between the ROC AUC results of Cone-OMP and OMP. White circles: OMP is better. Blue: Cone-OMP is better. Red: Cone-OMP is better in each of the 10 trials. Square: Cone-OMP is better than the best overall OMP result. An $\times$ symbol: best radius for the current n and s.

resentative error for that beat. The beats with the largest errors are considered abnormal. We partially borrowed the implementation from [18]; our code, written in MATLAB and not yet optimized, can be found at `http://asydil.upb.ro/software`.

We run our tests for $n \in \{64, 96, 128\}$, $s \in \{2, 3, 4, 5\}$. We take all the radii to be equal, and test with $\rho \in \{0.01, 0.02, \ldots, 0.1\}$. For each triplet (n, s, ρ), we generate 10 random dictionaries for each segment and, for each window, we compute 10 errors, as described above, then compute the average representative error for each beat. We also compute the average ROC AUC and the average number of false positives when the true positive rate is 100%, 97.5%, and 95%, corresponding to finding all 40 anomalies, 39, and 38, respectively.

A sample of the best values is given in Table 1. More illustrative on the detection power of Cone-OMP is the map shown in Figure 3, where ROC AUC values are compared. Colors show the radii for which Cone-OMP is better in average (blue) or, not only in average, but better for each of the 10 dictionaries (red). For many (n, s) pairs, there are several

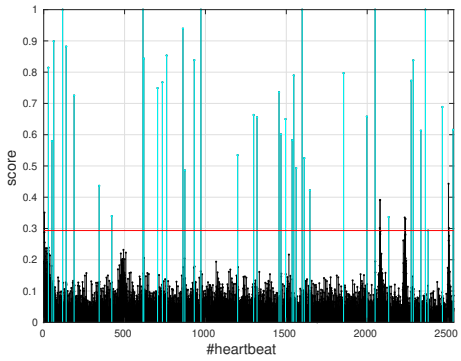

Fig. 4: Representative error for each heartbeat, computed with KSVD+OMP; black=normal, cyan=anomaly, red=smallest error of an anomaly.

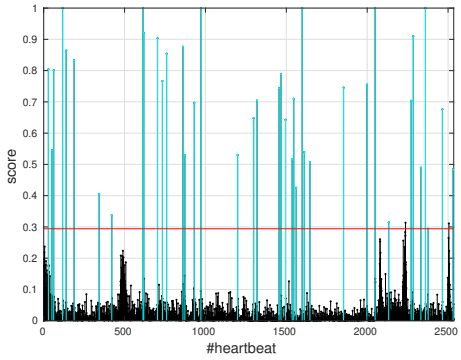

Fig. 5: Same as in Fig.4, for KSVD+Cone-OMP.

radii for which Cone-OMP gives a ROC AUC higher than the best overall OMP result (0.99971, see first line of Table 1). Cone-OMP tends to give better results for smaller values of the dictionary size. The behavior is robust with respect to the radius: there is a range of radii for which Cone-OMP is better. The results are clearly better than in [16].

To give some insight, Figures 4 and 5 show the representative error (normalized such that the highest value is 1) for OMP and Cone-OMP, respectively, for a single dictionary ($n = 96$, $s = 4$, $\rho = 0.04$). We see that Cone-OMP is able to represent very well the normal beats, the error being much closer to zero than for OMP. However, the abnormal beats still have large errors, showing that Cone-OMP has good potential for solving anomaly detection problems.

5. CONCLUSIONS AND FUTURE WORK

We have presented an OMP algorithm for sparse representations with cone atoms and its application to anomaly detection. Further work naturally extends to dictionary learning; we are currently testing several algorithms with promising results. A major challenge is the automated choice of the radii.

6. REFERENCES

[1] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, "A Survey of Sparse Representation: Algorithms and Applications," *IEEE Access*, vol. 3, pp. 490–530, 2015.

[2] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad, "Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition," in *27th Asilomar Conf. Signals Systems Computers*, Nov. 1993, vol. 1, pp. 40–44.

[3] G. Pope, C. Aubel, and C. Studer, "Learning phase-invariant dictionaries," in *Int. Conf. Acoustics Speech Signal Proc (ICASSP)*, Vancouver, Canada, May 2013, pp. 5979–5983.

[4] C. Rusu, B. Dumitrescu, and S.A. Tsaftaris, "Explicit shift-invariant dictionary learning," *IEEE Signal Proc. Letters*, vol. 24, no. 1, pp. 6–9, Jan. 2014.

[5] C. Garcia-Cardona and B. Wohlberg, "Convolutional dictionary learning: A comparative review and new algorithms," *IEEE Transactions on Computational Imaging*, vol. 4, no. 3, pp. 366–381, 2018.

[6] L. Le Magoarou and R. Gribonval, "Chasing Butterflies: in Search of Efficient Dictionaries," in *Int. Conf. Acoustics Speech Signal Proc (ICASSP)*, Brisbane, Australia, Apr. 2015, pp. 3287–3291.

[7] J. Song, X. Xie, G. Shi, and W. Dong, "Multi-layer discriminative dictionary learning with locality constraint for image classification," *Pattern Recognition*, vol. 91, pp. 135–146, 2019.

[8] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Proc.*, vol. 17, no. 1, pp. 53–69, 2008.

[9] S. Hawe, M. Seibert, and M. Kleinsteuber, "Separable dictionary learning," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 438–445.

[10] V.H. Nguyen, V.M. Patel, N.M. Nasrabadi, and R. Chellappa, "Design of Non-Linear Kernel Dictionaries for Object Recognition," *IEEE Trans. Image Proc.*, vol. 22, no. 12, pp. 5123–5135, Dec. 2013.

[11] A. Golts and M. Elad, "Linearized Kernel Dictionary Learning," *IEEE J. Sel. Topics Signal Proc.*, vol. 10, no. 4, pp. 726–739, June 2016.

[12] V. Papyan, Y. Romano, and M. Elad, "Convolutional neural networks analyzed via convolutional sparse coding," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 2887–2938, 2017.

[13] J. Hu and Y.-P. Tan, "Nonlinear dictionary learning with application to image classification," *Pattern Recognition*, vol. 75, pp. 282–291, 2018.

[14] S. Mahdizadehaghdam, A. Panahi, H. Krim, and L. Dai, "Deep dictionary learning: A parametric network approach," *IEEE Trans. Image Proc.*, vol. 28, no. 10, pp. 4790–4802, 2019.

[15] G.B. Moody and R.G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[16] A. Adler, M. Elad, Y. Hel-Or, and E. Rivlin, "Sparse coding with anomaly detection," *Journal of Signal Processing Systems*, vol. 79, no. 2, pp. 179–188, 2015.

[17] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Trans. Signal Proc.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[18] N. Cleju and I.B. Ciocoiu, "Preconditioned K-SVD for ECG anomaly detection," in *International Symposium on Electronics and Telecommunications (ISETC)*, Timisoara, Romania, 2020, pp. 1–4.