An In-depth Study of Sparse Codes on Abnormality Detection

Ren, Huamin; Pan, Hong; Olsen, Søren Ingvor; Jensen, Morten Bornø; Moeslund, Thomas B.

Published in:
IEEE Advanced Video and Signal-based Surveillance (AVSS) 2016

DOI (link to publication from Publisher):
10.1109/AVSS.2016.7738016

Publication date:
2016

Document Version
Publisher's PDF, also known as Version of record

An In-depth Study of Sparse Codes on Abnormality Detection

Huamin Ren
Aalborg University, Denmark
hr@create.aau.dk

Hong Pan, Søren Ingvor Olsen
University of Copenhagen, Denmark
hong.pan,ingvor@di.ku.dk

Morten Bornø Jensen, Thomas B. Moeslund
Aalborg University, Denmark
mboj,tbm@create.aau.dk

Abstract

Sparse representation has been applied successfully in abnormal event detection, in which the baseline is to learn a dictionary accompanied by sparse codes. While much emphasis is put on discriminative dictionary construction, there are no comparative studies of sparse codes regarding abnormality detection. We present an in-depth study of two types of sparse codes solutions - greedy algorithms and convex L1-norm solutions - and their impact on abnormality detection performance. We also propose our framework of combining sparse codes with different detection methods. Our comparative experiments are carried out from various angles to better understand the applicability of sparse codes, including computation time, reconstruction error, sparsity, detection accuracy, and their performance combining various detection methods. The experiment results show that combining OMP codes with maximum coordinate detection could achieve state-of-the-art performance on the UCSD dataset [14].

1. Introduction

Abnormal event detection is the core of video surveillance applications, which could assist people in various situations, such as monitoring patients/children, observing people and vehicles in a busy environment, or preventing theft and robbery. The purpose of abnormality detection is to learn normal patterns or behaviors through training and detect any abnormal or suspicious behaviors in test videos.

Abnormal event detection, like other video analysis applications such as action recognition, needs to address the following questions: how to represent image/video content effectively, how to learn patterns from the training data, and how to conduct the detection task. What differentiates abnormal event detection from other applications is that training videos only contain normal behaviors, and the determination of abnormal features is based on a detection method rather than a classification method. Sparse representation has been found beneficial due to the its compactness and representative ability, with which the remaining signals can be represented or reconstructed in terms of a linear combination of atoms in an overcomplete dictionary [19] [3] [8]. The use of sparse representation has also turned out to be successful in other applications, such as image denoising [5] and action recognition [11].

Research on sparse representation can be generally divided into dictionary learning [13] and sparse coding [15] [18] [2] [10]. Dictionary learning aims to obtain atoms (or basis vectors) for a dictionary. Such atoms could be either predefined, e.g., undecimated Wavelets [9], steerable Wavelets, Contourlets [4], Curvelets [1], and more variants of Wavelets, or learned from the data itself, such as the K-SVD [16] [7] method and the BSD algorithm [19]. Sparse coding, on the other hand, attempts to find sparse codes (or coefficients) with a given dictionary, i.e., finding the solution to the underdetermined system of equations $y = Dx$ either by greedy algorithms or convex algorithms. Through sparse coding, input features can be approximately represented as a weighted linear combination of a small number of (unknown) basis vectors. These methods include matching pursuit [15], orthogonal matching pursuit [18], and basis pursuit [2].

When applying sparse representation on abnormal event detection, a great deal of emphasis is put on dictionary learning. A common procedure is: first, visual features are extracted either on a spatial or temporal domain. A dictionary D is then learned based on these visual features, which consists of basis vectors capturing high-level patterns in the input features, as in [13, 19]. A sparse representation of a feature is a linear combination of a few elements or atoms from a dictionary. Mathematically, this can be expressed as $y = Dx$, where $y \in R^p$ is a feature of interest, $D \in R^{p\times m}$ is a dictionary, and $x \in R^m$ is the sparse representation of

978-1-5090-3811-4/16/$31.00 ©2016 IEEE

y in D. Typically $m \gg p$ results in an overcomplete or redundant dictionary. During the detection procedure, each testing feature can be determined as normal or an anomaly based on its reconstruction error.

Despite the aforementioned progress, the impacts of sparse codes that serve as important feature representations to distinguish normalities from anomalies remain to be explored. For instance, among the huge body research on coding representations, which performs well in detecting anomalies? For a given sparse code, most of the existing approaches use only an approximate reconstruction error to save computation; for example, the least square error, which means the sparse codes are actually not taken into consideration during the detection. In fact, the impact of sparse codes generated by different approaches is still unclear. Therefore, we offer an in-depth study of the sparse codes in terms of their performance on abnormal event detection. We pay special attention to two major types of sparse codes: greedy algorithms and L1-norm minimization algorithms.

Our main contributions are: 1) we provide an in-depth study of sparse codes, in terms of their reconstruction error, sparsity, computation time and detection performance on anomaly datasets; 2) we propose a framework to detect abnormality, which combines sparse representation with various detection methods; and 3) we propose an effective code detection method, Maximum Coordinate (MC), which achieves the state-of-the-art performance on the UCSD anomaly detection dataset [14] when combining with the OMP codes [18].

The remainder of this paper is organized as follows: We give a brief review of greedy algorithms and L1-norm solutions in Sec.2 and propose our framework of abnormal event detection in Sec.3, which combines sparse codes with various detection methods. We show our comparative results in Sec.4 and conclude the paper with a discussion and ideas of future work in Sec.5.

2. Sparse Codes Representation

There are various ways of generating sparse codes through optimization solutions. We introduce two categorized solutions: greedy algorithms and L1-norm approximation solutions.

Greedy algorithms rely on an interactive approximation of the feature coefficients and supports, either by iteratively identifying the support of the feature until a convergence criterion is met, or by obtaining an improved estimate of the sparse signal at each iteration that attempts to account for the mismatch with the measured data. Compared to L1-norm minimization methods, greedy algorithms are much faster, and thus are more applicable to very large problems.

L1-norm minimization, on the other hand, has become a popular tool to solve sparse coding, which benefits both from efficient algorithms and a well-developed theory for generalization properties and variable selection consistency [21]. We list two common L1-norm minimization formulations in E.q. 1 and E.q. 2. Since the problem is convex, there are efficient and accurate numerical solvers.

$$\hat{x} = \operatorname*{argmin}_{x} \frac{1}{2}\|Dx - y\|_2^2 + \lambda\|x\|_1 \tag{1}$$

$$\hat{x} = \operatorname*{argmin}_{x} \|x\|_1 \quad \text{subject to} \quad \|Dx - y\|_2 \leq \epsilon \tag{2}$$

2.1. Greedy Algorithms

We review two broad categories of greedy methods to reconstruct y, which are called 'greedy pursuits' and 'threshold' algorithms. Greedy pursuits can be defined as a set of methods that iteratively build up an estimate x. They contains three basic steps. First, the x is set to a zero vector. Second, these methods estimate a set of non-zero components of x by iteratively adding new components that are deemed to be non-zeros. Third, the values for all non-zeros components are optimized. In contrast, thresholding algorithms alternate between element selection and element pruning steps.

There is a large and growing family of greedy pursuit methods. The general framework in greedy pursuit techniques is 1) to select an element and 2) to update the coefficients. Matching Pursuit (MP) [15] discusses a general method for approximate decomposition in E.q. 3, which addresses the sparsity issue directly. The algorithm selects one column from D at a time and only the coefficient associated with the selected column is updated at each iteration. More concretely, it starts from an initial approximation x (0) = 0 and residual $R(0) = x$, then builds up to a sequence of sparse approximations stepwise. At stage k, it identifies the dictionary atom that best correlates with the residual and then adds to the current approximation a scalar multiple of that atom. After m steps, one has a sparse code, seen in E.q. 3 with residual $R = R^{(m)}$.

$$y = \sum_{i=1}^{m} x_{r_i} d_{r_i} + R^{(m)} \tag{3}$$

Orthogonal Matching Pursuit (OMP) [18], updates x in each iteration by projecting y orthogonally onto the columns of D associated with the current support atoms. In contrast to MP, OMP never reselects an atom and the residual at any iteration is always orthogonal to all currently selected atoms in the dictionary. Another difference between the two is that OMP minimizes the coefficients for all selected atoms at iteration k, while MP only updates the coefficient of the most recently selected atom. In order to speed up pursuit algorithms, it is necessary to select

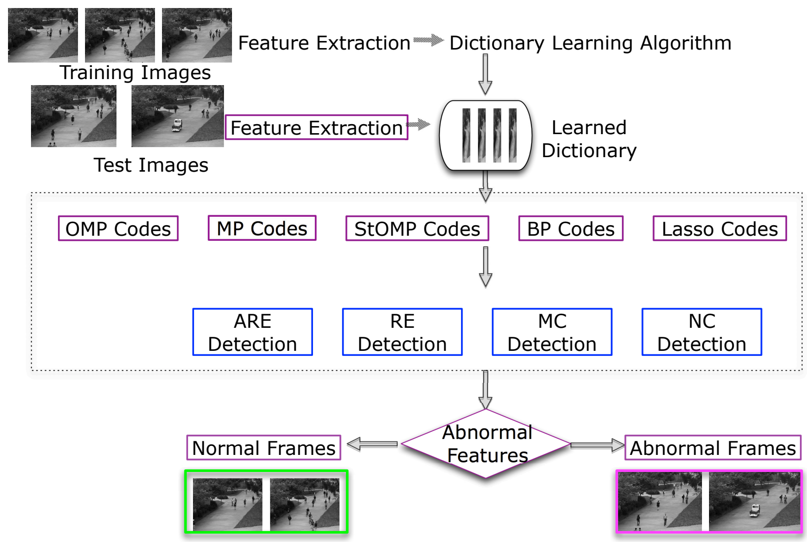

Figure 1. Our framework of combining sparse codes with different detection methods.

multiple atoms at a time; therefore, the algorithms are proposed to keep computational costs low enough for applying to large-scale problems, such as Stagewise Orthogonal Matching Pursuit (StOMP) [6]. These algorithms choose the element that meets some threshold criterion at the atom selection step and have demonstrated both theoretical and empirical effectiveness for the large system.

Greedy algorithms are easy to implement and use and can be extremely fast. However, they do not have recovery guarantees, i.e., how well each sample can be reconstructed by the dictionary and their sparse codes, in contrast to L1-norm approximations.

2.2. L1-norm Approximation

L1-norm approximation replaces the L0 constraint with a relaxed L1-norm. For example, in the Basis Pursuit method (BP) [2], a nearly universally differentiable and often convex cost function is applied, while in the Focal Underdetermined System Solver (FOCUSS) algorithm [17], a more general model is optimized.

Donoho and etc. [5] suggest that for some measurement matrices D, the generally NP-Hard problem (L0 norm) should be equivalent to its convex relaxation: L1 norm, see E.q. 1. The convex L1 problem can be solved using methods of linear programming. Representative work includes Basis Pursuit (BP). Instead of seeking sparse representations directly, it seeks representations that minimize the L1 norm of the coefficients. Furthermore, BP can compute sparse solutions in situations where greedy algorithms fail. The Lasso algorithm [20] is similar to BP and is, in fact, know as Basis Pursuit De-Noising (BPDN) in some areas. Rather than trying to minimize the L1-norm like BP, the Lasso places a restriction on its value.

The FOCUSS algorithm has two integral parts: a low-resolution initial estimate of the real signal and the iteration process that refines the initial estimate to the final localized energy solution. The iterations are based on the weighted norm minimization of the dependent variable for which the weights act as a function of the preceding iterative solutions. The algorithm is presented as a general estimation tool that can be used across various applications. In general, L1-norm methods offer better performance in many cases, but they are also more demanding with respect to computation.

3. Sparse Code Based Detection

In addressing the detection of abnormal behaviors based on sparse codes, two issues should be addressed: 1) how to generate the sparse codes, i.e., the solution of x; and 2) how to determine whether the testing code is normal or anomalous. For the first issue, various sparse codes discussed in Sec.2 could be adopted, while for the second issue we take various detection methods into consideration. Our proposed abnormal event detection framework is shown in Fig. 1.

After a testing feature is represented by a sparse code, the detection method determines whether it is normal or abnormal. There are two commonly used detection methods: the reconstruction error (RE) and the approximated reconstruction error (ARE). In terms of sparse codes, the high response of dictionary atoms, or concentrated non-zeros in coefficients, may indicate a connection to a possible normality. Unfortunately, these codes property and their connection with normality or abnormality have not been explored yet. Therefore, we also introduce maximum coordinate (MC) and the non-zero concentration (NC) as two new detection methods.

Reconstruction Error (RE): Most existing approaches treat dictionary learning and detection as two separate processes, i.e., a dictionary is typically learned based on the training data, and then different measurements are adopted to determine whether the testing sample is an anomaly. More sophisticated approaches unify these two processes into a mixed reconstructive and discriminative formulation. Nevertheless, a basic measurement that is widely used in both cases is reconstruction error. The reconstruction error of the testing sample y, according to the dictionary D, is represented as: $\|y - D\alpha\|_2^2$, where α is the sparse code of y.

Approximate Reconstruction Error (ARE): To speed up detection, reconstruction error is sometimes approximated by the least squares [13] rather than being calculated based on sparse codes through an optimization solution. Thus, the reconstruction error is calculated as: $\|y - D(D^TD)^{-1}D^Ty\|$.

Maximum Coordinate (MC): Given a testing sample y, its sparse code is denoted as α. Ideally, all non-zero entries in the estimate α would be associated with the columns of the dictionary from a normal pattern (note that only normal data is used during the training). Then we could detect y as a normal feature if a single largest entry in α were found; otherwise, it would be detected as an anomaly.

Non-zero Concentration (NC): Inspired by [19], the distribution of non-zeros is more important to the detection than the location of non-zero elements. Thus, we propose a detection measurement called non-zero concentration. Based on the dictionary proposed in [19], a normal code should have a non-zero concentration property, i.e., non-zeros concentrated in the dictionary that has the smallest reconstruction error. Anomalies can be detected if no concentration is found on any of the existing dictionaries.

4. Experimental Results

We provide a comprehensive study of the abnormality detection performance on sparse codes. Our experiments are carried out on the UCSD [14] Ped1 dataset because it is a popular abnormal event detection dataset and many detection results are reported. We start by evaluating the performance of various sparse codes, focusing on a comparison of sparse codes generated by two types of algorithms: greedy algorithms and L1-norm approximation algorithms. The following aspects are highlighted: computation time, reconstruction error, the ratio of sparsity in codes, and their performance on abnormal event detection. Next, we use the OMP algorithm to generate sparse codes, combine the codes with different detection methods, and conclude by evaluating their detection performance with state-of-the-art algorithms.

4.1. Dataset and Settings

UCSD Ped1 dataset [14] is a frequently used public dataset for detecting abnormal behaviors. It includes clips of groups of people walking towards and away from the camera with some perspective distortion. There are 34 training videos and 36 testing videos with a resolution of 238 $\times$ 158. Training videos contain only normal behaviors. Testing videos demonstrate abnormal behaviors exhibited by either non-pedestrian entities in the walkways or anomalous pedestrian motion patterns.

We use the spatial-temporal cubes in which 3D gradient features are computed, which mimic the setting in [12]. Each frame is divided into patches of a size of 23 $\times$ 15. Five consecutive frames are used to form 3D patches, and gradients features are extracted in each patch. See details in [12]. Through this, we obtain 500-dimensional visual features and reduce them to 100 dimension by using the PCA algorithm.

4.2. Comparison of Sparse Codes

We evaluate sparse codes from four perspectives: computation time, reconstruction error, the ratio of sparsity in codes, and the codes' performance on abnormal event detection based on their reconstruction error.

We randomly select 1% of the training features (238,000 features in total), use the K-SVD algorithm [16] to construct a dictionary consisting of 1000 atoms, and generate sparse codes by applying various algorithms. There are many algorithms available; we select only representative greedy algorithms (OMP, MP, StOMP) and compare them with representative L1-norm solutions (BP and Lasso algorithm). The reconstruction error is calculated by $Re = \|y - Dx\|_2^2$. We also calculate the mean ratio of sparsity in the codes, i.e., the average percentage of non zeros in the dimension of the codes (1000). We report these results as well as computation time in Tab. 1. Greedy algorithms need far less time to compute, and the OMP achieves the fastest computation, followed by the StOMP algorithm. OMP is approx. 180 times faster than the Lasso algorithm. Both OMP and StOMP could achieve sparser solutions, while BP could obtain an extremely dense solution with an exact recovery.

To measure the accuracy of abnormality detection, we calculate the reconstruction error of each feature and register features with large reconstruction errors as anomalies. A frame with an abnormal feature is considered a positive frame. To compare performance, we adopt two popular evaluation criteria in abnormality detection: frame-level evaluation and pixel-level evaluation, which are defined in [14]. We follow precisely their setting in our evaluation, which is to say that in the frame-level evaluation, a frame is considered abnormal if it contains at least one

Table 1. Comparison of greedy algorithms and L1-norm solutions on sparse code generation.

Algorithms	Computation Time (s)	Reconstruction Error	Sparsity (%)
MP	166.00	0	31.8%
OMP	1.83	0.4236	1.9%
StOMP	15.79	0	10%
BP	114.20	0	100%
Lasso	333.49	0.0005	9.9%

Table 2. Comparative results on UCSD Ped1: frame-level evaluation results (AUC and EER) and pixel-level evaluation results (AUC and EDR) are reported.

Algorithms	AUC (frame-level)	EER	AUC (pixel-level)	EDR	Computation Time (s)
MP	0.6956	0.3547	0.3898	0.5716	13342
OMP	0.5003	0.5052	0.2849	0.6637	527
StOMP	0.5415	0.465	0.3494	0.6190	4668
BP	0.5454	0.4764	0.3057	0.6479	38949
Lasso	0.5305	0.5173	0.3132	0.6383	56400

anomaly feature. In contrast, for the pixel-level evaluation, a frame is marked as a correctly detected abnormality if at least 40% of the truly abnormal pixels are detected. Ground truth on frame-level and pixel-level annotation is available, and we calculate the true positive and false positive rates to draw ROC curves, and report the Area Under the Curve (AUC). Following [14], we obtain the value when the false positive number equals the missing value. These are called the equal error rate (EER) and equal detected rate (EDR) in the frame and pixel-level evaluations, respectively. See Tab. 2 for details. In the frame-level evaluation, the MP algorithm achieves the best results with a moderate computation time. The StOMP algorithm is relatively fast, and the AUC is satisfactory.

It is worth noting that the pixel-level AUC is lower than the frame-level AUC in general because the pixel-level evaluation is stricter and takes location into consideration. In the frame-level evaluation, there could be a coincidental detection - a normal feature could be erroneously detected as an anomaly in an abnormal frame, and this erroneous detection could end up with a correct detection of that frame. In the pixel-level evaluation, in contrast, a frame is marked as a correctly detected abnormality only if a sufficient number of anomaly features has been found. Compared to the MP algorithm, the StOMP algorithm can achieve a competitive detection result in the pixel-level evaluation, but it is three times faster than the MP algorithm. The BP algorithm also performs well on pixel-level detection; however, its high computation cost hampers its application in real detection problems.

In summary, greedy algorithms compute quickly, but their reconstruction errors are larger than L1-norm solutions. Convex relaxations, such as the BP and the Lasso algorithm, have better theoretical guarantees and recovery ability, but they are more time consuming. Surprisingly, greedy algorithms, especially the StOMP algorithm, seem to perform better on pixel-level detection, which means that they could more accurately localize anomaly features.

4.3. Comparison of Combining Sparse Codes with Detection Methods

We choose the OMP algorithm to generate sparse codes due to computation considerations, combine them with four types of detection methods (RE, ARE, MC, NC), and compare their detection performance. We draw comparative frame-level AUC curves that correspond to the detection methods. We then compare these combinations with state-of-the-art methods on abnormality detection.

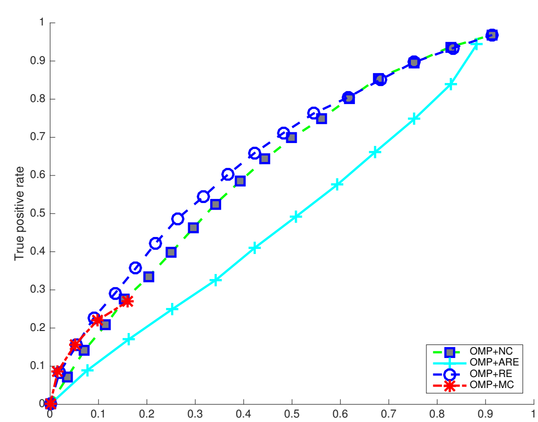

Figure 2. Combining sparse codes generated by the OMP algorithm with various detection methods: non-zero concentration (NC), approximate reconstruction error (ARE), reconstruction error (RE), and maximum coordinate (MC).

Table 3. Comparative values of AUC, EER and EDR on UCSD Ped1 dataset.

Method	AUC (frame-level)	EER	AUC (pixel-level)	EDR
SF-MPPCA [14]	0.5900	0.3200	0.2130	0.3200
MDT [14]	0.8180	0.2500	0.4610	0.2500
Lu13[13]	0.5842	0.4413	0.3622	0.5826
OMP+RE	**0.6603**	**0.3823**	0.5386	0.5113
OMP+ARE	0.5013	0.5081	0.5317	0.5113
OMP+NC	0.5697	0.5055	0.5397	0.5113
OMP+MC	0.6339	0.4016	**0.5433**	**0.5113**

As displayed in Fig. 2, abnormality detection by computing the real reconstruction error outperforms the estimated reconstruction error on frame-level evaluation, which further validates the idea that the decomposition of real coefficients is necessary. Among all approaches, OMP+RE achieves the best AUC score on frame-level evaluation (0.6603), followed by MC (0.6340), NC (0.5697) and ARE (0.5013). We give further insight into how accurate the detection is in an even stricter pixel-level evaluation. We find that OMP+MC achieves the best result with an AUC of 0.5433. This is because that the high response in the code means that there is a strong connection between the testing feature with some atoms in the dictionary. This happens when the features have a similar pattern to the atoms convey. Therefore, the high response also implies that the testing feature is normal. However, we also notice that NC detection, which also considers the non-zeros distribution in sparse codes, performs relative poorly. This may be due to the type of dictionary being adopted, or due to the principle of how the OMP code is generated, which are based on the reconstruction error of the chosen atoms, rather than the concentrated atoms.

Finally we compare combining OMP codes and various detection methods with state-of-the-art abnormality detection algorithms. Comparison of AUC in the frame-level evaluation of UCSD Ped1 is shown in Fig. 3, and quantized evaluations are shown in Tab. 3. Compared with state-of-the-art algorithms, combining OMP codes with detection methods outperform other methods on two criteria evaluation except MDT method on frame-level evaluation, which verifies the effectiveness of sparse codes generated by greedy algorithms. Note that as aforementioned pixel-level is a more precisely defined evaluation criterion, even though the AUC of MDT [14] on frame-level evaluation is higher than OMP codes, its AUC on pixel-level evaluation is low, which may indicate a high false positive prediction of abnormal pixels. Furthermore, maximum coordinate detection outperforms other methods, which implies that a high response (large code value) could contribute to the detection.

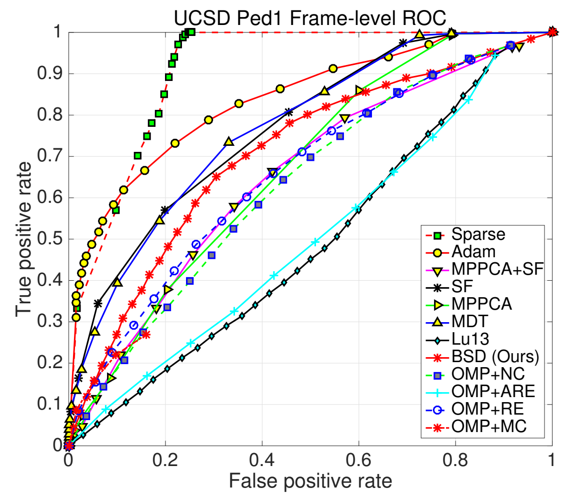

Figure 3. Comparison with state-of-the-art abnormality detection approaches.

5. Discussions and Conclusion

In this paper, we give an in-depth study of sparse codes in respect to their performance in abnormal event detection. We compare two category sparse codes: codes generated by greedy algorithms and those generated by L1-norm solutions. Various aspects are covered: computational cost, recovery ability, sparsity, and their detection performance. Furthermore, we explore the sparse codes and compare different methods to determine whether a testing code is an anomaly or not.

Experimental results show that greedy algorithms can obtain good detection results with fewer computations. Among the top three best detection results, two are greedy algorithms. Considering the computation requirement, which limits some L1-norm algorithms from being applied in real surveillance applications, greedy algorithms are promising. When combining OMP codes with various detection measurements, maximum coordinate measurement outperforms other methods, which implies that the high response in the code could help the detection result.

References

[1] E. J. Cands and D. L. Donoho. Recovering edges in ill-posed inverse problems: optimality of curvelet frames. *The Annals of Statistics*, 30(3):784–842, 2002.

[2] S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998.

[3] Y. Cong, J. Yuan, and J. Liu. Sparse reconstruction cost for abnormal event detection. In *CVPR*, pages 3449–3456, 2011.

[4] M. N. Do and M. Vetterli. Contourlets. In *Beyond Wavelets*. Academic Press, 2003.

[5] D. L. Donoho and M. Elad. *Optimally Sparse Representation in General (non-orthogonal) Dictionaries Via L1 Minimization*. Department of Statistics, Stanford University, 2002.

[6] D. L. Donoho, Y. Tsaig, I. Drori, and J. L. Starck. Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 58(2):1094–1121, 2012.

[7] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15:3736–3745, 2006.

[8] S. R. Fanello, I. Gori, G. Metta, and G. Odone. Keep it simple and sparse: Real-time action recognition. *Journal of Machine Learning Research*, 14:2617–2640, 2013.

[9] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *PAMI*, 13(9):891–906, 1991.

[10] K. Huang and S. Aviyente. Sparse representation for signal classification. In *In Adv. NIPS*, 2006.

[11] Z. Jiang, Z. Lin, and L. S. Davis. Label consistent k-svd: Learning a discriminative dictionary for recognition. *PAMI*, 35(11):2651–2664, 2013.

[12] L. Kratz and K. Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *CVPR*, pages 1446–1453, 2009.

[13] C. Lu, J. Shi, and J. Jia. Abnormal event detection at 150 fps in matlab. In *ICCV*, pages 2720–2727, 2013.

[14] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *CVPR*, pages 1975–1981, 2010.

[15] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12):3397–3415, Dec 1993.

[16] A. Michal, E. Michael, and B. Alfred. K-svd: Design of dictionaries for sparse representation. In *SPARS'05*, pages 9–12, 2005.

[17] J. Murray and K. Kreutz-Delgado. An improved focuss-based learning algorithm for solving sparse linear inverse problems. In *Signals, Systems and Computers*, volume 1, pages 347–351 vol.1, Nov 2001.

[18] Y. C. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers*, pages 40–44 vol.1, Nov 1993.

[19] H. Ren, W. Liu, S. E. S. Olsen, and T. B. Moeslund. Unsupervised behavior-specific dictionary learning for abnormal event detection. In *BMVC*, 2015.

[20] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.

[21] T. Zhang. Some sharp performance bounds for least squares regression with l1 regularization. *Ann. Statist.*, 37:2109–2144, 2009.