



Feature Synthesization for Real-Time Pedestrian Detection in Urban Environment

Wenhua Fang¹(✉), Jun Chen¹, Tao Lu², and Ruimin Hu¹

¹ National Engineering Research Center for Multimedia Software, Computer School of Wuhan University, Wuhan 430072, Hubei Province, China
{fangwh, chenj, hrm}@whu.edu.cn

² Computer School of Wuhan Institute of Technology, Wuhan 430205, Hubei Province, China
lutxyl@gmail.com

Abstract. Real-time pedestrian detection is very essential for auto assisted driving system. For improving the accuracy, more and more complicate features are proposed. However, most of them are impracticable for the real-world application because of high computation complexity and memory consumption, especially for onboard embedding system in the unmanned vehicle. In this paper, a novel framework that utilizes reconstruction sparsity to synthesize the feature map online is proposed for real-time pedestrian detection for the early warning system of the unmanned vehicle in real world. In this framework, the feature map is computed by sparse line combination of the representative coefficient and the feature response of trained basis which is learned offline. The efficiency of our method only depends on the dictionary decomposition no matter how complicated the feature is. Moreover, our method is suitable for most of the known complicate features. Experiments on four challenging datasets: Caltech, INRIA, ETH and TUD-Brussels, demonstrate that our proposed method is much efficient (more than 10 times acceleration) than the state-of-the-art approaches with comparable accuracy.

Keywords: Pedestrian detection · Feature synthesization
Sparse representation

1 Introduction

Recently, the unmanned vehicle, as a new transportation which has the merits of energy saving and environmental protection, is getting more and more attentions. Meanwhile, its security is the focus of the debate. As we all know, obstacles identification is one of the core functions of the early warning system of unmanned vehicle, and how to detect pedestrian as soon as possible in real world is one of the key problems in obstacle recognition.

Pedestrian detection is a very important task in computer vision and has great potential to apply in many fields, such as automatic assisted driving, intelligent traffic management, etc. It is very challenging because of the multiple views, different illuminations, multiple scales and partial occlusion. To overcome these problems, many researchers have proposed a lot of complicated features [1–7] to improve the accuracy of this task, but ignored the efficiency. Take the popular object detector [4] for example. It will take more than 4 s per image with the resolution of 352×288 and take even more than 30 s per image with the resolution of 1280×720 on the 4-core desktop computer. Nowadays with the wide application of high resolution cameras, this problem is more and more serious and has become a bottleneck for real-time application.

From the common framework of pedestrian detection, we can see that the time cost is proportional to the product of two parts. One is the time complexity of the detector, and the other is the time for probing the object candidates. For the exhaustive search, such as sliding window technique, there are usually tens of thousands of probing candidates for pedestrian classification and location. So many researchers have carried out the work in the above two aspects to improve the efficiency of the pedestrian detection [8–12]. Felzenszwalb et al. [9] proposed a cascade part pruning strategy to speed up the deformable part model by more than ten times. In [8], Yan et al leveraged the low rank constraint on root filter to get a 2D correlation between root filter and feature map, and used the lookup table to speed up the HOG extraction. And it was 4 times faster than the current fastest DPM method with similar accuracy. Besides, many other efforts, called region proposal, have been done to get the object location candidate prior to object detection. In [10], Sande et al proposed a hierarchical framework, named *selective search*, to generate approximate 2000 regions per image by color segmentation and the recall was up to 97%. Compared with the exhaustive search, it was much effective. However, due to the computation complexity of selective search, it is not very fit for the real-time application. Zitnick [11] proposed a method, named *Edgebox*, to gain the region candidates at a lightweight computational cost, but its recall rate is relatively low. Other approaches are also facing similar problems. Recently, deep neural network, named deep learning, has become the state-of-the-art approach in object detection [13, 14]. But this kind of methods are too complex to need auxiliary computation equipment, such as GPU, to complete the long-term training and testing.

From the above analysis, we can see that the time consumption of feature extraction is the key for efficient pedestrian detection. Is there an approach that the time consumption is approximately fixed for most of features? In this paper, we proposed a framework based on sparse coding to conduct the feature extraction online by linear combination of features of dictionary atoms extracted offline. This is based on the assumption that the natural image can be linearly combined by the patches sparsely. If the feature satisfies the linear superposition principle, the feature extraction can be synthesized online and the time consumption is just decided by the image decomposition.

2 Related Work

Our work is inspired by [15, 16]. The core idea in [15] is the shared representations. And an intermediate representation, called *sparselet*, for deformable part models was proposed for multi-class object detection. In this model, sparse coding of part filters was used to represent each filter as a sparse linear combination of shared dictionary elements, which are the parameters of the part filter. Reconstruction of the original part filter responses via sparse matrix-vector product reduces computation relative to conventional part filter convolutions. The main defect is the sacrifice of the performance. In [16], *sparselet* was reformulated in a general structured output prediction framework leading to larger speedup factors with no decrease in task performance. We think more deeply about the problem. Compared with them, our model has smaller granularity and is more general. Our main contribution is that we first consider the feature response synthesization for feature extraction for any pedestrian detection framework by sparse coding. And we demonstrate that the synthesized feature has comparable performance with fast feature extraction for online pedestrian detection.

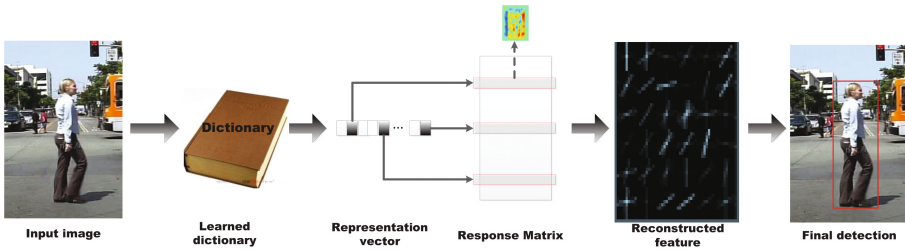


Fig. 1. The framework of feature response synthesization for pedestrian detection.

3 Our Framework

In this section, we will introduce the framework in details. Our model is based on the fact that natural image can be represented by linear combination of redundant bases. There are two stages (offline training and online detecting) in our framework. According to standard sparse representation, we first learn a representative dictionary in the training dataset under the minimum reconstruction error and sparsity constraint. And then the response matrix is created by conducting the feature extraction operation on the items of the dictionary (the item is regarded as patches). On the detecting stage, an input image is represented by the sparse representative vector on the learned dictionary. In the vector, only a few items are non-zero. And then the feature response synthesization is regarded as the linear combination of the representative vector and the row items of the response matrix (shown as Fig. 1).

What conditions should be satisfied if the feature is fit for our framework? We think that the linear superposition principle should be satisfied. The principle is as follows:

$$f(ax + by) = af(x) + bf(y) \quad (1)$$

where $f(*)$ denotes the feature extraction operation. Under certain conditions, we can relax the restriction of linear superposition. For example, if $f(ax) = a^n f(x)$ is satisfied, the feature is still fit for our method.

How many features are fit for our methods? According to the above constraints, most of popular features used in pedestrian detection are all suitable, such as HOG [1], LBP [6], ICF [17], ACF [18]. Because the HOG feature is the basis of some complex approaches, such as DPM [4], our method can accelerate many other complex pedestrian detection frameworks which include the above features.

3.1 Region Proposal

In object detection task, for locating the object, traditional methods scan the image using multiple windows with different scales in the zigzag manner, named sliding window strategy, and then discriminate whether it include an object or not in each window. Usually, it will probe more than one million times. So such exhaustive search strategy is not fit for our real-time applications because it is very time-consuming. After analysis of such method, we can observe that a large proportion of probing is in the background. So if the background regions before scanning can be excluded, the efficiency of the detection will be boosted in a large margin. Recently, many efforts are made to generate the object candidates (bounding boxes) for object detection, called region proposal [10, 24, 25]. Because the decomposition of the whole image based on the trained dictionary is much time-consuming, the region proposal is critical for real-time object detection. After in-depth investigation [24], we take the Edgebox [25] as our region proposal approach, because it is most efficient under the highest recall rate.

3.2 Dictionary Learning

Given a set of image patches $Y = [y_1, \dots, y_n]$, the standard unsupervised dictionary learning algorithm aims to jointly find a dictionary $D = [d_1, \dots, d_m]$ and an associated sparse code matrix $X = [x_1, \dots, x_n]$ by minimizing the reconstruction error as follows.

$$\min_{D, X} \|Y - DX\|_F^2 \quad s.t. \forall i, \|x_i\|_0 \leq K \quad (2)$$

where x_i are columns of X , the zero-norm $\|\cdot\|_0$ counts the non-zero entries in the sparse code x_i and K is a predefined sparsity level.

Although the above optimization is NP-hard, greedy algorithms such as orthogonal matching pursuit algorithm (OMP) [19, 26, 27] can be used to efficiently compute an approximate solution. In our experiment, we use K-SVD

algorithm [19] to train the discriminative dictionary. In addition, we consider three sparsity inducing regularizers:

- (1) Lasso Penalty [28]

$$R_{Lasso}(a) = \lambda_1 \|a\|_1$$

- (2) Elastic net penalty [29]

$$R_{EN}(a) = \lambda_1 \|a\|_1 + \lambda_2 \|a\|_2^2$$

These regularizers lead to convex optimization problems, and employ a two step process to get the solution. In the first step, a subset of the activation coefficients is selected to satisfy the constraint $\|a\|_0 \leq \lambda_0$. In the second step, the selection of nonzero variables is fixed (thus satisfying the sparsity constraint) and the resulting convex optimization problem is solved.

3.3 Feature Response Synthesization

Feature response synthesization can be regarded as the linear combination of the representative coefficients and the response of the items of the learned dictionary. Denoting the feature pyramid of an image I as Φ , and $I = [P_1, \dots, P_N]$, and D_j in $D = [D_1, \dots, D_K]$ is the atom of D (Dictionary), we have $\Psi * P_i \approx \Psi * (\sum_j \alpha_{ij} D_j) = \sum_j \alpha_{ij} (\Psi * D_j)$, where $*$ denotes the convolution operator. Concretely, we can recover individual part filter responses via sparse matrix multiplication (or lookups) with the activation vector replacing the heavy convolution operation as shown in Eq. 3.

$$\begin{bmatrix} \Psi * P_1 \\ \Psi * P_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \Psi * P_N \end{bmatrix} \approx \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \alpha_N \end{bmatrix} \begin{bmatrix} \Psi * D_1 \\ \Psi * D_2 \\ \vdots \\ \Psi * D_K \end{bmatrix} = AM \quad (3)$$

For efficient pedestrian detection, the extraction of some features should be made appropriate adjustments. Take HOG feature for example. It is composed of concatenated blocks. Each block includes 2×2 cells, and each cell is the 8×8 pixels of the image. So the block is 16×16 pixels. The concatenation of histograms of the blocks has two strategies: overlap and non-overlap. In the overlap manner, the sliding step width is usually the width of the cell. In the non-overlap manner, the sliding step width is the width of the block. So the dimension of the feature of the non-overlap is smaller than that of the overlap. But the performance of the feature will be lost by nearly 1% [1]. So the standard HOG feature chooses the overlap manner for better performance. For high acceleration, in this paper, we choose the non-overlap manner.

4 Experiments

For evaluating our method, we conduct the experiments on four challenging pedestrian datasets: Caltech [20], INRIA [1], ETH [21] and TUD-Brussels [22]. The state-of-the-art and classic pedestrian detectors are chosen to test our framework: HOG [1], ChnFtrs [5], ACF [18], HOGLBP [7], LatSvmV2 [4] and Very-Fast [23]. In the experiments, the training and testing data setting is as same as in [18]. We first discuss the relation of the performance versus the sparsity degree, the size of the dictionary, the size of atom. And then we evaluate our method.

4.1 Dictionary Learning vs Performance

Because our method is based on sparse coding, how to select the parameters of dictionary learning directly affects the performance of feature reconstruction. For choosing the best parameters, we conduct some experiments on INRIA Person Dataset and the type of synthesized feature is HOG. INRIA Person Dataset consists of 1208 positive training images (and their reflections) of standing people, cropped and normalized to 64×128 , as well as 1218 negative images and 741 test images. This dataset is an ideal setting, as it is what HOG was designed and optimized for, and training is straightforward.

Sparsity Level and Dictionary Size. Figure 3 shows the average precision on INRIA when we change the sparsity level along with the dictionary size using 5×5 patches. We observe that when the dictionary size is small, a patch cannot be well represented with a single codeword. However, when the dictionary size grows and includes more structures in its codes, the $K = 1$ curve catches up, and performs very well. Therefore we use $K = 1$ in all the following experiments.

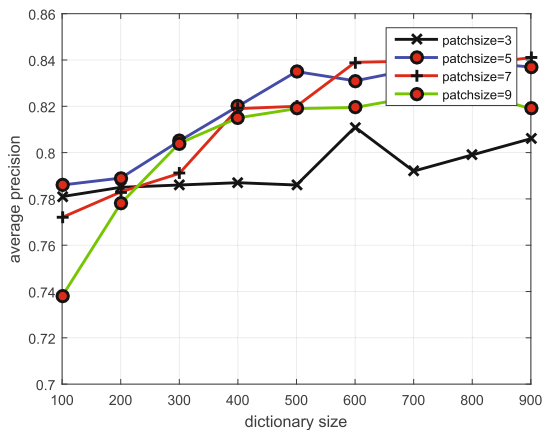


Fig. 2. The patch size vs detection performance on Caltech pedestrian dataset.

Patch Size and Dictionary Size. Next we investigate whether our synthesized features can capture richer structures using larger patches. Figure 2 shows the average precision as we change both the patch size and the dictionary size. While 3×3 codes barely show an edge, 7×7 codes work much better. However, 9×9 patches, may be too large for our setting and do not perform well.

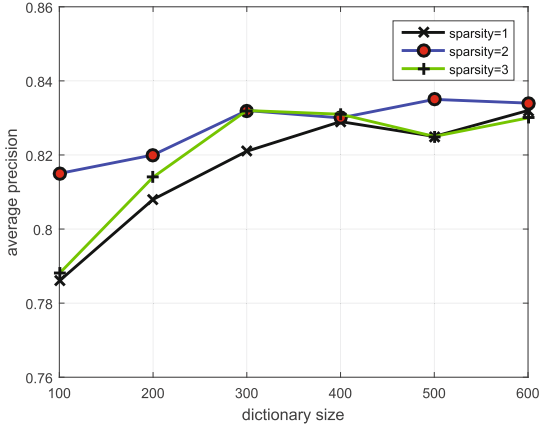


Fig. 3. The sparsity vs detection performance on Caltech pedestrian dataset.

Regularizer. With $K = 1$, one can also use different regularizers to learn a dictionary. Figure 4 compares the detection accuracy with Lasso penalty vs Elastic net penalty on 7×7 patches. The Elastic net penalty is better because it include more constraints to learn discriminative representation.

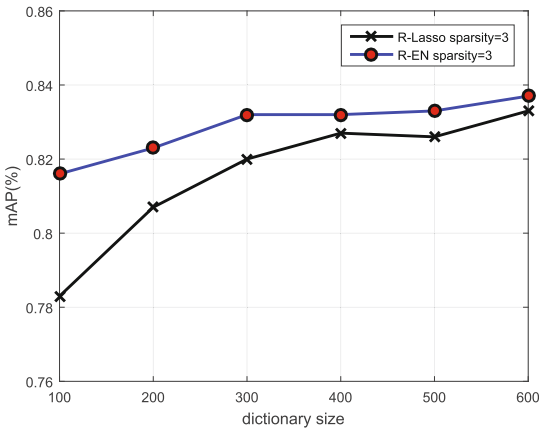


Fig. 4. The regularizer vs detection performance on Caltech pedestrian dataset.

In the following experiments, we set the size of the dictionary, the sparsity degree to be 600, 1 respectively. We set the size of the atom of the dictionary to be 7×7 for better performance.

4.2 Performance Comparison

We just pay attention to whether the performance is lost and the degree of performance loss. As shown in Table 1, we can see the performance comparison between the original detector and the corresponding synthesized detector. As can be seen from the table, the performance degradation is very small, about one percent. Why is the synthesis method a little worse than the original method? We think there are at least two reasons. One is that our method is based on reconstruction error minimum and sparsity constraints, which cause the loss of the discriminative information for pedestrian detection. The other is that we slightly modified the original feature extraction, such as HOG in the non-overlap manner. Compared to high speedup, we think this slight performance degradation is worth.

Table 1. Missing rates of pedestrian detectors (origin vs synthesizer) on four challenging datasets. “origin” denotes the original detector, and “synthesizer” stands for the synthesized feature by our framework.

Detectors	INRIA [1]		Caltech [20]		TUD-Brussels [22]		ETH [21]	
	Origin	Synthesizer	Origin	Synthesizer	Origin	Synthesizer	Origin	Synthesizer
HOG [1]	46	48	68	69	78	78	64	65
ChnFtrs [5]	22	23	56	57	60	61	57	58
ACF [18]	17	18	43	45	50	51	50	52
HOGLBP [7]	39	40	68	68	82	83	55	56
LatSvmV2 [4]	20	23	63	64	70	71	51	52
VeryFast [23]	16	17	53	54	61	62	55	56

4.3 Speed Comparison

The speed of the detector is more important than performance in the real-world applications. In this section, we will show the speed comparison of the above origin detectors and the corresponding synthesized detectors. Because the speed of the detector depends on the resolution.

We just do the statistics and analysis on the INRIA dataset because the results on the other datasets are the same as that on this dataset. The resolution of the image is 640×480 in INRIA testing set. As shown in Table 2, acceleration of the synthesized detector is very obvious. Take the detector HOGLBP [7] for example. The speedup ratio is up to 2000. The speed of original veryfast detector [23] is 50 fps because it is accelerated by GPU. But the speed of our synthesized detector is 110 fps. From the table, experiment results confirm our conjecture that the runtime of our synthesized detector depends on the decomposition of the image based on the dictionary.

Table 2. Speed comparison of pedestrian detectors (origin vs synthesizer) on INRIA person dataset. The unit of speed is the frame of per second.

Detectors	INRIA [1]	
	Origin	Synthesizer
HOG [1]	0.23	96.5
ChnFtrs [5]	16.4	121.2
ACF [18]	31.9	125.4
HOGLBP [7]	0.06	120.4
LatSvmV2 [4]	0.6	108.5
VeryFast [23]	50	110.2

5 Conclusion

In this paper, we proposed a novel framework of feature extraction based on sparse representation. And we give the constraint condition that the feature should satisfy in our framework. At last, we conduct enough experiments on four challenging datasets to evaluate our method. Experiment results demonstrate our method is efficient for pedestrian detection task. In the future, we will seek the efficient dictionary learning method and consider to add the classification error into dictionary learning to add the distinctive information.

Acknowledgment. This research is based upon work supported by National Nature Science Foundation of China (No. U1736206), National Nature Science Foundation of China (61671336), National Nature Science Foundation of China (61671332), Technology Research 10 F. Author et al. Program of Ministry of Public Security (No. 2016JJSYJA12), Hubei Province Technological Innovation Major Project (No. 2016AAA015), Hubei Province Technological Innovation Major Project (2017AAA123), The National Key Research and Development Program of China (No.2016YFB0100901), Nature Science Foundation of Jiangsu Province (No. BK20160386) and National Nature Science Foundation of China (61502354).

References

1. Navneet, D., Bill, T.: Histograms of oriented gradients for human detection. In: Proceedings of the 22nd IEEE Conference on Computer Vision and Pattern Recognition, pp. 886–893, June 2005
2. Sabzmejdani, P., Greg, M.: Detecting pedestrians by learning shapelet features. In: Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, June 2007
3. Bo, W., Ram, N.: Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In: Proceedings of the Tenth IEEE International Conference on Computer Vision, pp. 90–97, June 2005
4. Pedro, F., David, M., et al.: A discriminatively trained, multiscale, deformable part model. In: Proceedings of the 25th IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, June 2008

5. Piotr, D., Zhuowen, T., et al.: Integral channel features. In: Proceedings of the 20th British Machine Vision Conference, pp. 250–258, September 2009
6. Ahonen, T., Hadid, A., Pietikäinen, M.: Face recognition with local binary patterns. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3021, pp. 469–481. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24670-1_36
7. Xiaoyu, W., Tony, X., et al.: An HOG-LBP human detector with partial occlusion handling. In: Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, pp. 32–39, June 2009
8. Junjie, Y., Zhen, L., et al.: The fastest deformable part model for object detection. In: Proceedings of the 32nd IEEE Conference on Computer Vision and Pattern Recognition, pp. 2497–2504, June 2014
9. Pedro, F., Ross, B., et al.: Cascade object detection with deformable part models. In: Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition, pp. 2241–2248, June 2010
10. Uijlings, J. R., Van De Sande, K. E., et al.: Selective search for object recognition. *Int. J. Comput. Vis.* **104**(2), 154–171 (2013)
11. Zitnick, C.L., Dollár, P.: Edge boxes: locating object proposals from edges. In: Proceedings of the 18th European Conference on Computer Vision, pp. 391–405, September 2014
12. Cheng, M.M., Zhang, Z., et al.: Bing: binarized normed gradients for objectness estimation at 300 fps. In: Proceedings of the 32nd IEEE Conference on Computer Vision and Pattern Recognition, pp. 3286–3293, June 2014
13. Ren, S., He, K., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings of the 33rd IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1–8, June 2015
14. Li, J., Liang, X., et al.: Scale-aware fast R-CNN for pedestrian detection. *Comput. Sci.* 25–32 (2015)
15. Song, H.O., et al.: Sparselet models for efficient multiclass object detection. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, pp. 802–815. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33709-3_57
16. Girshick, R., Song, H.O., et al.: Discriminatively activated sparselets. In: Proceedings of the 30th International Conference on Machine Learning, pp. 196–204, June 2013
17. Dollr, P., Tu, Z., et al.: Integral channel features. In: Proceedings of the 20th British Machine Vision Conference, pp. 7–10, September 2009
18. Dollar, P., Appel, R., et al.: Fast feature pyramids for object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(8), 1532–1545, May 2014
19. Aharon, M., Elad, M., et al.: K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.* **54**(11), 4311–4322, October 2006
20. Dollar, P., Wojek, C., et al.: Pedestrian detection: an evaluation of the state of the art. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**(4), 743–761, March 2011
21. Andreas, E., Bastian, L., et al.: Depth and appearance for mobile scene analysis. In: Proceedings of the 25th IEEE International Conference on Computer Vision, pp. 1–8, June 2007
22. Wojek, C., Walk, S., et al.: Multi-cue onboard pedestrian detection. In: Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, pp. 794–801, June 2009

23. Benenson, R., Mathias, M., et al.: Pedestrian detection at 100 frames per second. In: Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition, pp. 2903–2910, June 2012
24. Hosang, J., Benenson, R., Dollar, P., et al.: What makes for effective detection proposals? *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(4), 814–830, March 2015
25. Zitnick, C.L., Dollr, P.: Edge boxes: locating object proposals from edges. In: Proceedings of the 18th European Conference on Computer Vision, pp. 391–405, September 2014
26. Cotter, S.F., Rao, B.D., et al.: Forward sequential algorithms for best basis selection. *IEEE Vis. Image Signal Process.* **146**(5), 235 (1999)
27. Mallat, S.G., Zhang, Z.: Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.* **41**(12), 3397–3415 (1993)
28. Tibshirani, R.: Regression shrinkage and selection via the lasso. *J. Roy. Stat. Soc.* **58**(1), 267–288 (1996)
29. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. *J. Roy. Stat. Soc.* **67**(2), 301–320 (2005)