# BLIND IMAGE QUALITY ASSESSMENT WITH A PROBABILISTIC QUALITY REPRESENTATION

*Hui Zeng*[1], *Lei Zhang*[1], *Alan C. Bovik*[2]

[1]Department of Computing, The Hong Kong Polytechnic University.
[2]Department of Electrical and Computer Engineering, The University of Texas at Austin.

## ABSTRACT

Most existing blind image quality assessment (BIQA) methods learn a regression model to predict scalar quality scores. Such a scheme ignores the fact that an image will receive divergent subjective scores from different subjects, which cannot be adequately represented by a single scalar number. This is particularly true on complex, real-world distorted images. However, the more informative score distributions are unavailable in existing image quality assessment (IQA) databases and can be potentially noisy when limited number of opinions are collected on each image. This paper proposes a probabilistic quality representation (PQR) and employs a more robust loss function to train deep BIQA models. Using a very straightforward implementation, the proposed method is shown to not only speed up the convergence of deep model training, but also greatly improve the quality prediction accuracy relative to scalar quality score regression methods under the same setting. The source code is available at `https://github.com/HuiZeng/BIQA_Toolbox`.

***Index Terms*—** Image quality assessment, image quality representation, score distribution

## 1. INTRODUCTION

Blind image quality assessment (BIQA) remains a very challenging problem due to the unavailability of a reference image. Most existing BIQA methods follow the flowchart shown in Fig. 1. Classical BIQA methods [1, 2, 3, 4, 5] typically first extract some handcrafted features (e.g., derived from natural scene statistics models) to represent the distorted image, and then train a regression model (e.g., by support vector regression (SVR)) to map the feature representations to subjective quality scores. Deep learning based BIQA methods have been attracting increasing attention in recent years because of the remarkable capability of deep neural networks to learn discriminative features. These deep methods generally train a convolutional neural network (CNN) to regress the scalar quality score in an end-to-end manner. Although several attempts [6, 7, 8, 9, 10] have been made, it remains a difficult task to train a robust deep BIQA model because of the very limited number of training samples on which human subjective scores have been collected [11].

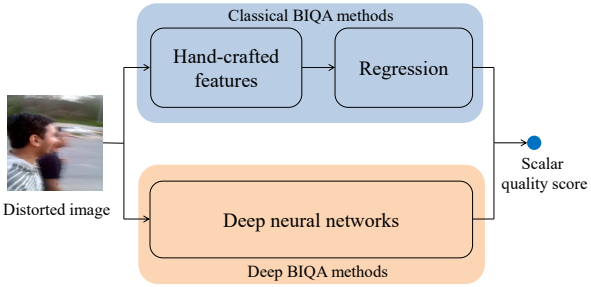



**Fig. 1**. Flowchart of existing BIQA methods. Both the classical and the deep BIQA methods directly train a model to regress the scalar quality scores.

A common limitation of existing BIQA methods is that only the scalar Mean Opinion Score (MOS) is used to train a regression model. Such a scheme ignores the fact that an image will receive divergent subjective scores from different human raters. This is particularly true on complex, real-word distorted images. Specifically, the average standard deviation of the subjective scores of the images in the LIVE Challenge (hereafter referred to as the LIVE-C) database [12] is 19.27 on the MOS scale of [0,100]. The subjective property may not be adequately represented by a single scalar number, and potentially useful and predictive information contained in the distributions of subjective scores has been rarely discussed or utilized in the literature. Unfortunately, the more informative score distributions are unavailable in existing image quality assessment (IQA) databases and can be statistically unreliable when limited number of opinions are collected on each image.

This paper proposes a probabilistic quality representation (PQR) to approximately describe the subjective score distribution of each image. Based on this new representation, we are able to employ a more robust loss function to train deep BIQA models. Our method not only speeds up the training process, but also achieves much higher prediction accuracy than widely used scalar quality score regression scheme under exactly the same setting.

## 2. PROBABILISTIC QUALITY REPRESENTATION

The framework of training a CNN model using the proposed PQR is shown in Fig. 2. The PQR is consisted of three key components: the quality anchors, probabilistic representation mapping and the reverse mapping to scalar scores.

**Quality anchors.** Given a training set with $N$ samples $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$, where $y_n$ is the MOS of the $n$-th training sample $\mathbf{x}_n$, the first step is to find $M$ quality anchors that fall within the overall range of quality scores. A straightforward strategy is to divide the numerical range of possible subjective scores into a small number of equally spaced intervals. For example, one could partition the subjective quality range into five Likert-type levels representing "bad," "poor," "fair," "good," and "excellent" [13, 12]. These natural divisions may already be available as part of the subjective dataset being used. We divide the score range, typically [0, 1] or [0, 100], into $M$ equal bins, then define the midpoints $\{c^m\}_{m=1}^M$ of the bins as the quality anchors. For simplicity, $M$ is fixed at 5.

**Probabilistic representation.** The scalar quality scores are then mapped into vectorized PQRs; that is, each image is assigned a set of probabilities of the quality anchors. The PQR is designed under the following two constraints: 1) Given an image with assigned MOS $y_n$, define a set of probabilities $q_n^m$ associated with the anchors $\{c^m\}_{m=1}^M$, such that $q_n^m$ is large when the Euclidean distance $\|y_n - c^m\|^2$ is small, and decreases monotonically with increasing distance; 2) The per-image anchor probabilities sum to 1. A simple and effective function is the soft-mapping function:

$$q_n^m = g(y_n, m) = \frac{\exp(-\beta\|y_n - c^m\|^2)}{\sum_{i=1}^M \exp(-\beta\|y_n - c^i\|^2)}, \quad (1)$$

where $q_n^m$ is the probability that the $n$-th training sample belongs to the $m$-th quality anchor, and $\beta$ is a scaling constant. In our implementations, we normalized $y_n$ to the range [0,1] on all examined databases, and determined a common value of $\beta = 64$ over all databases via cross-validation.

**Reverse mapping.** By transforming the scalar image quality score into a PQR vector, a deep BIQA model is learned which will output PQR vectors descriptive of perceptual image quality. Since the basic evaluation criteria of modern BIQA models are generally computed on the scalar MOS, it is desirable to re-map the output PQR vectors back to scalar quality scores. This is a vector-to-scalar transform problem, which we solve by learning a regression function $h(\cdot)$ that maps PQR vectors back to scalar quality scores, specifically, by minimizing the following error function:

$$err = \frac{1}{N}\sum_{n=1}^N \|h(\mathbf{q}_n) - y_n\|^2, \quad (2)$$

where $\mathbf{q}_n$ is the vector form of $\{q_n^m\}_{m=1}^M$. This reverse mapping is a relatively simple regression task that can be easily and accurately solved using a linear SVR model. We have found the average absolute error $\frac{1}{N}\sum_{n=1}^N |h(\mathbf{q}_n) - y_n|$ of this minimization to be smaller than 0.01 on a MOS scale of [0, 1]) for reasonable choices of $\beta$ and $M$.

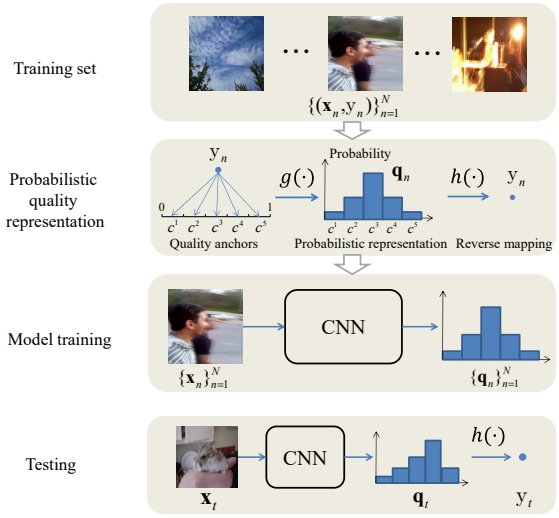



**Fig. 2**. Framework of training a CNN model using the PRQ.

**Loss function.** Considering that the transformed probabilities $\mathbf{q}_n$ lie in the range $[0, 1]$ and sum to 1, we employ a softmax layer to ensure that the output of the deep model satisfies the same properties. Denote by $\widetilde{\mathbf{q}}_n$ the output of the softmax layer. Since both the output of the deep model $\widetilde{\mathbf{q}}_n$ and the target $\mathbf{q}_n$ are probability distributions, it is a natural and effective choice to minimize the Kullback-Leibler (KL) divergence between these two probability distributions:

$$D_{KL}(\mathbf{q}_n\|\widetilde{\mathbf{q}}_n) = \sum_{m=1}^M q_n^m \log\frac{q_n^m}{\widetilde{q}_n^m}. \quad (3)$$

Since the target probability distribution $\mathbf{q}_n$ is fixed, minimizing the KL-divergence is identical to minimizing the cross-entropy [14, Chap. 6.9]. Our final loss function is then:

$$\min_\omega \frac{1}{N}\sum_{n=1}^N\sum_{m=1}^M -q_n^m \log\widetilde{q}_n^m. \quad (4)$$

Training deep BIQA models using our proposed PQR exploits some attractive optimization properties compared to traditional scalar quality score regression using the mean square error (MSE). Firstly, by introducing domain knowledge, the PQR supplies much richer information descriptive of the subjective opinions of the training samples and enforces stronger constraints on the highly flexible deep CNN models, thereby stabilizing the training process and supplying better generalization capability. Moreover, since the PQR is in a probabilistic form, it is natural to employ the softmax cross-entropy loss, which yields much faster convergence than the MSE loss [15]. During the testing stage, the PQR prediction vector of each sample is re-mapped to a scalar quality score using the learned function $h(\cdot)$ in Eq. 2.

## 3. EXPERIMENTS

### 3.1. Experimental Setup

Four representative IQA databases including LIVE-C [12], LIVE [13], CSIQ [16] and TID2013 [17] were employed in our experiments. Three different CNN models, including the pre-trained AlexNet and ResNet50, and a shallow CNN model (hereafter referred to as S_CNN) with 5 convolutional layers and 0.9 million parameters were evaluated. When fine tuning the pre-trained AlexNet and ResNet50 models, we randomly extracted 50 image crops (of sizes $227 \times 227$ for AlexNet and $224 \times 224$ for ResNet50) from each training image, except on TID2013, we extracted 25 crops per image since this database contains more distorted images. All of the image crops inherited the PQR of the source image. The fine-tuning process iterated for 20 epochs, using a batch size of 256 for AlexNet, and 10 epochs with a batch size of 64 for ResNet50. The learning rate was set to be a logarithmically spaced vector in the interval [1e-3,1e-4] for both models. When training the S_CNN from scratch, we extracted 500 image patches (of size $64 \times 64$) per image on each database (except 250 patches on TID2013). The network parameters of S_CNN were randomly initialized and the training process was allowed to iterate for 40 epochs using a batch size of 1024. The learning rate was set to be a logarithmically spaced vector in the interval [1e-2,1e-3].

In the testing stage, we extracted overlapped image patches at a fixed stride (64 for the fine tuned deep models and 32 for the shallow S_CNN model) from each test image. The PQR prediction vector of each image crop was mapped to a scalar quality score using the pre-trained linear SVR model. Average pooling was used to output a final whole-image quality score. Two metrics, Spearman's rank correlation coefficient (SRCC) and Pearson's linear correlation coefficient (PLCC), were used to evaluate the performances of the learned BIQA models. On each database, we randomly divided the samples into a training set with 80% images and a testing set with 20% images, without overlap in image content. All the experiments were repeated 10 times and the median SRCC and PLCC were reported as the final results. The MatConvNet toolbox [18] was used to train the CNN models on a PC equipped with a GTX 1080Ti. The SVR model was trained using the LIBSVM toolbox [19].

### 3.2. Comparison Against Scalar Regression

We first compared the performances of deep BIQA models trained on PQR against commensurate models trained on traditional scalar quality representation (SQR) in regards to both convergence speed and prediction accuracy. For fair comparison, all settings except for the loss function were held constant across all models. All the four databases and the three CNN models were evaluated. The median SRCC and PLCC across 10 repetitions are reported for each combination in Table 1. The learning curves obtained on LIVE-C are also shown in Fig. 3, where the standard deviations of S_CNN are not plotted to make the curves more distinguishable.

The following observations can be made from Table 1 and Fig. 3. First, the BIQA models trained using our proposed PQR model significantly outperform the BIQA models trained by the traditional SQR model on all the three CNN architectures. This convincingly shows that our proposed PQR model is a very effective new tool for deep BIQA model learning. Secondly, under the same settings, BIQA model training using the PQR model converges much faster than the SQR model, especially in regards to fine tuning the deep models, where we found that the PQR-based model converges within no more than 3 epochs. Finally, our method results in a much smaller standard deviation of prediction performance, which strongly suggests that the probabilistic representation is much more robust and stable than directly regressing the deterministic scalar quality scores using the MSE loss.

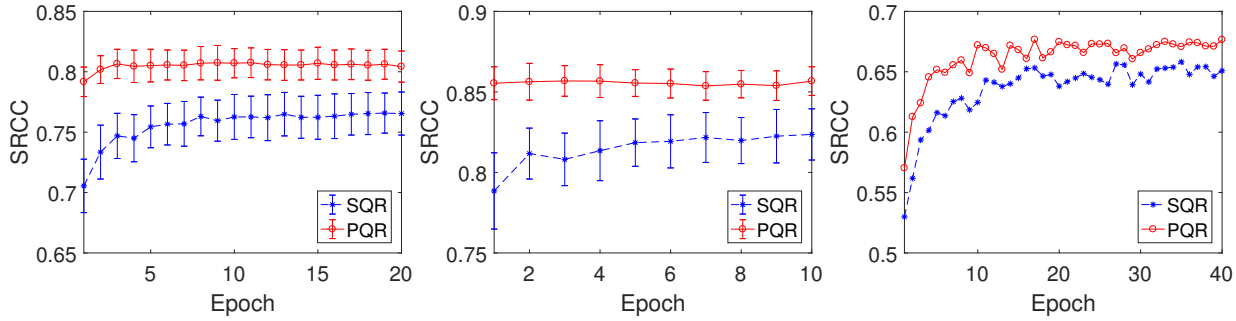
### 3.3. Comparison Among Different CNN Models

A very interesting observation can be made on Table 1 that: the S_CNN trained from scratch can achieve performance that is competitive, or even superior to that achieved by the two pre-trained deep CNN models on the three legacy databases, but much worse than the two deep models on the authentic LIVE-C database. This divergence in performance can likely be explained in terms of the different characteristics of the databases. The legacy databases contain a limited variety of synthetic distortion types and degradation levels, which have been homogeneously applied in isolation to a small number of source images. Because of this, the mapping from perceptual quality degradation to quality scores is relatively easy to learn, even by a shallow CNN model. Moreover, the greater degree of spatial distortion homogeneity makes it possible to leverage small image patches when training a shallow CNN model, because they are more representative of the distortions afflicting the whole image. However, the LIVE-C database contains many highly diverse contents that are authentically distorted in many complex combinations and degrees. These complex, real-world multi-distortions are often quite inhomogeneous. Thus the LIVE-C is simply too complex for a shallow CNN model to be able to achieve good performance.

On the other hand, deep CNN models pre-trained on a target problem like the ImageNet classification task can generalize well to other image recognition and processing tasks [20]. However, the transferability of a network depends on the degrees of statistical similarity between the training data and target data [21]. Since both ImageNet and LIVE-C consist of natural images afflicted with authentic distortions (although with no overlap and from very different sources), it is natural to infer that models pre-trained on ImageNet may effectively transfer to the quality prediction task on the LIVE-C database. By contrast, the three legacy databases are composed of images algorithmically modified by synthetic distortions. It fol-

| CNN model | Representation | LIVE-C | | LIVE | | CSIQ | | TID2013 | |
|---|---|---|---|---|---|---|---|---|---|
| | | SRCC | PLCC | SRCC | PLCC | SRCC | PLCC | SRCC | PLCC |
| AlexNet | SQR | 0.7658 | 0.8074 | 0.9319 | 0.9462 | 0.7965 | 0.8405 | 0.5362 | 0.6136 |
| | PQR | 0.8075 | 0.8357 | 0.9554 | 0.9638 | 0.8713 | 0.8958 | 0.5742 | 0.6687 |
| ResNet50 | SQR | 0.8236 | 0.8680 | 0.9468 | 0.9527 | 0.8217 | 0.8713 | 0.6406 | 0.7068 |
| | PQR | **0.8568** | **0.8822** | **0.9653** | **0.9714** | 0.8728 | 0.9010 | **0.7399** | **0.7980** |
| S_CNN | SQR | 0.6582 | 0.6729 | 0.9450 | 0.9455 | 0.8787 | 0.8987 | 0.6526 | 0.6921 |
| | PQR | 0.6766 | 0.7032 | 0.9637 | 0.9656 | **0.9080** | **0.9267** | 0.6921 | 0.7497 |

**Table 1**. Comparing the best performance of PQR against SQR using different CNN models on different databases.




(a) AlexNet (b) ResNet50 (c) S_CNN

**Fig. 3**. Learning curves of three CNN models on the LIVE-C database.

| Methods | SRCC | PLCC |
|---|---|---|
| DIIVINE [1] | $0.58 \pm 0.03$ | $0.60 \pm 0.03$ |
| CORNIA [2] | $0.63 \pm 0.04$ | $0.66 \pm 0.04$ |
| BRISQUE [3] | $0.61 \pm 0.03$ | $0.65 \pm 0.04$ |
| NIQE [22] | $0.43 \pm 0.03$ | $0.48 \pm 0.03$ |
| HOSA [8] | $0.66 \pm 0.04$ | $0.68 \pm 0.03$ |
| FRIQUEE-ALL [5] | $0.69 \pm 0.03$ | $0.71 \pm 0.03$ |
| Bosse *et al.* [10]$^\star$ | 0.67 | 0.68 |
| PQR (AlexNet) | $0.81 \pm 0.01$ | $0.84 \pm 0.01$ |
| PQR (ResNet50) | $\mathbf{0.86 \pm 0.01}$ | $\mathbf{0.88 \pm 0.01}$ |
| PQR (S_CNN) | $0.68 \pm 0.03$ | $0.70 \pm 0.03$ |

**Table 2**. Comparisons with existing BIQA models in LIVE-C. The results marked by $^\star$ is copied from the original paper.

lows that the statistics of the images in these databases are very different from those in ImageNet, hence the transferability of models pre-trained on ImageNet is greatly reduced.

### 3.4. Comparison with the state-of-the-art on LIVE-C

Finally, we also compared the proposed PQR-based models against existing BIQA methods on the LIVE-C database. Since the split of training and testing sets will affect prediction accuracy, for fair comparison, we re-ran the source codes of the DIIVINE [1], CORNIA [2], BRISQUE [3], NIQE [22], HOSA [8] and FRIQUEE-ALL [5] using the same training and testing splits as we used for the PQR model. We tried our best to optimize the corresponding parameters of each method on each database. The implementation details of these methods can also be found in our released code. The median SRCC and PLCC over 10 random rounds are reported in Table 2.

From Table 2, it may be observed that our probabilistic deep BIQA models achieve standout results among all of the competing methods. When using ResNet50, it outperformed the previous best results (obtained by FRIQUEE-ALL) by more than 0.15 in regards to both SRCC and PLCC. This again demonstrates that fine tuning pre-trained deep models using the PQR is an effective way to improve predictions of the perceptual quality of authentically distorted images.

## 4. CONCLUSION

We were able to train deep BIQA models using our probabilistic quality representation (PQR) to accurately predict image quality, while achieving faster convergence with a greater degree of stability. By imposing probabilistic constraints on the learned prediction mapping, we essentially regularize the learning process. Our extensive experiments on existing IQA databases demonstrated that deep models trained using PQR were able to achieve promising quality prediction accuracy, especially on the LIVE-C database, which is composed of real-world images degraded by complex, authentic distortions.

## 5. ACKNOWLEDGEMENT

This work is supported by China NSFC grant (no. 61672446).

## 6. REFERENCES


[1] A. K. Moorthy and A. C. Bovik, "Blind image quality assessment: from natural scene statistics to perceptual quality," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3350–3364, 2011.

[2] P. Ye, J. Kumar, L. Kang, and D. Doermann, "Unsupervised feature learning framework for no-reference image quality assessment," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1098–1105.

[3] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, 2012.

[4] W. Xue, X. Mou, L. Zhang, A. C. Bovik, and X. Feng, "Blind image quality assessment using joint statistics of gradient magnitude and laplacian features," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4850–4862, 2014.

[5] D. Ghadiyaram and A. C. Bovik, "Perceptual quality prediction on authentically distorted images using a bag of features approach," *Journal of Vision*, vol. 17, no. 1, pp. 1–25, 2017.

[6] L. Kang, P. Ye, Y. Li, and D. Doermann, "Convolutional neural networks for no-reference image quality assessment," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1733–1740.

[7] H. Tang, N. Joshi, and A. Kapoor, "Blind image quality assessment using semi-supervised rectifier networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2877–2884.

[8] W. Hou, X. Gao, D. Tao, and X. Li, "Blind image quality assessment via deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 6, pp. 1275–1286, 2015.

[9] J. Kim and S. Lee, "Fully deep blind image quality predictor," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 1, pp. 206–220, 2017.

[10] S. Bosse, D. Maniry, K.-R. Müller, T. Wiegand, and W. Samek, "Deep neural networks for no-reference and full-reference image quality assessment," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 206–219, 2018.

[11] J. Kim, H. Zeng, D. Ghadiyaram, S. Lee, L. Zhang, and A. C. Bovik, "Deep convolutional neural models for picture-quality prediction: Challenges and solutions to data-driven image quality assessment," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 130–141, 2017.

[12] D. Ghadiyaram and A. C. Bovik, "Massive online crowdsourced study of subjective and objective picture quality," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 372–387, 2016.

[13] H. R. Sheikh, Z. Wang, L. Cormack, and A. C. Bovik, "Live image quality assessment database release 2," 2005.

[14] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[15] P. Golik, P. Doetsch, and H. Ney, "Cross-entropy vs. squared error training: a theoretical and experimental comparison." in *Interspeech*, vol. 13, 2013, pp. 1756–1760.

[16] E. C. Larson and D. M. Chandler, "Most apparent distortion: full-reference image quality assessment and the role of strategy," *Journal of Electronic Imaging*, vol. 19, no. 1, pp. 011 006–011 006, 2010.

[17] N. Ponomarenko, O. Ieremeiev, V. Lukin, K. Egiazarian, L. Jin, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti *et al.*, "Color image database TID2013: Peculiarities and preliminary results," in *Proc. 4th IEEE Eur. Workshop Vis. Inf. Process. (EUVIP),*, 2013, pp. 106–111.

[18] A. Vedaldi and K. Lenc, "MatConvNet – convolutional neural networks for MATLAB," in *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.

[19] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[20] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806–813.

[21] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems*, 2014, pp. 3320–3328.

[22] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a completely blind image quality analyzer," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, 2013.